



**TUGAS AKHIR - TE145561**

**PERANCANGAN *INVERSE KINEMATICS* ROBOT  
*MANIPULATOR 4 DEGREE OF FREEDOM***

Adi Novitarini Putri  
NRP. 2214030010

Dosen Pembimbing  
Slamet Budiprayitno, ST.,MT.  
Yunafi'atul Aniroh, ST., MSc.

PROGRAM STUDI KOMPUTER KONTROL  
Departemen Teknik Elektro Otomasi  
Fakultas Vokasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2017

*Halaman ini sengaja dikosongkan*



**FINAL PROJECT - TE145561**

***DESIGN INVERSE KINEMATICS ROBOT MANIPULATOR  
4 DEGREE OF FREEDOM***

Adi Novitarini Putri  
NRP. 2214030010

*Supervisor*  
Slamet Budiprayitno, ST., MT.  
Yunafi'atul Aniroh, ST., MSc.

**COMPUTER CONTROL STUDY PROGRAM**  
*Electrical and Automation Engineering Department*  
*Vocational Faculty*  
*Sepuluh Nopember Insitute of Technology*  
Surabaya 2017

*Halaman ini sengaja dikosongkan*

## PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul :

***“Perancangan Inverse Kinematics Robot Manipulator 4 Degree Of Freedom”***

adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 19 Juli 2017



Adi Novitarini Putri  
NRP. 2214030010

*Halaman ini sengaja dikosongkan*

**PERANCANGAN INVERSE KINEMATICS ROBOT MANIPULATOR 4  
DEGREE OF FREEDOM**

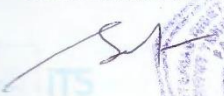
**TUGAS AKHIR**

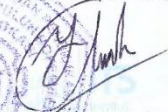
Diajukan Guna Memenuhi Sebagian Persyaratan  
Untuk Memperoleh Gelar Ahli Madya  
Pada  
Program Studi Komputer Kontrol  
Departemen Teknik Elektro Otomasi  
Fakultas Vokasi  
Institut Teknologi Sepuluh Nopember

Menyetujui :

Dosen Pembimbing I

Dosen Pembimbing II

  
Slamet Budiprayitno, S.T., M.T.

  
Yunafizatul Aniroh, S.T., M.Sc

NIP. 197811132010121002 NIP. 2200201405003

**SURABAYA  
JULI, 2017**

*Halaman ini sengaja dikosongkan*



# **PERANCANGAN INVERSE KINEMATICS ROBOT MANIPULATOR 4 DEGREE OF FREEDOM**

Adi Novitarini Putri  
2214030010

Dosen Pembimbing I : Slamet Budiprayitno, ST., M.T.  
NIP : 197811132010121002  
Dosen Pembimbing II : Yunafi'atul Aniroh, ST., M.Sc.  
NIP : 2200201405003

## **ABSTRAK**

Robot *manipulator* merupakan suatu sistem mekanik yang digunakan untuk meringankan kerja manusia. Ketika memindahkan benda dari satu titik ke titik yang lain, masing-masing *joint* pada robot *manipulator* harus bergerak dengan sudut yang sesuai agar dapat mencapai posisi yang diinginkan. Pada penelitian ini membahas mengenai pergerakan robot *manipulator 4 degree of freedom* dimana robot yang digunakan memiliki satu untaian kinematika yang terdiri dari 4 *joints* dengan tipe *revolute*. Sudut pada masing-masing *joint* dapat ditentukan dengan menggunakan sebuah metode yang digunakan dalam penelitian ini adalah metode *inverse kinematics*. Dari hasil penelitian dilakukan pengujian terhadap metode *inverse kinematics* dimana terdapat nilai kesalahan dari perbandingan hasil perhitungan dengan pengukuran sebenarnya. Nilai kesalahan untuk koordinat X, Y, dan Z adalah 4.556%, 14.818%, dan 6.2515%.

**Kata Kunci** *inverse kinematics*, robot *manipulator*

*Halaman ini sengaja dikosongkan*

# **DESIGN INVERSE KINEMATICS ROBOT MANIPULATOR 4 DEGREE OF FREEDOM**

Adi Novitarini Putri  
2214030010

Dosen Pembimbing I : Slamet Budiprayitno, ST., M.T.  
NIP : 197811132010121002  
Dosen Pembimbing II : Yunafi'atul Aniroh, ST., M.Sc.  
NIP : 2200201405003

## **ABSTRACT**

*Robot manipulator is a mechanical system used to alleviate human labor. When moving objects from one point to another, each joint on the robot manipulator must move at the correct angle to achieve the desired position. In this study discussed the movement of robot manipulator 4 degree of freedom where the robot used has a kinematic chains consisting of 4 joints with type revolute. The angle on each joint can be determined using a method used in this study is the inverse kinematics method. From the results of research conducted tests on inverse kinematics method where there is an error value of the comparison of the calculation results with the actual measurement. The error values for X, Y, and Z coordinates are 4.556%, 14.818%, and 6.2515%.*

**Keywords:** *DH-Parameter, inverse kinematics, robot manipulator*

*Halaman ini sengaja dikosongkan*

## KATA PENGANTAR

Puji dan syukur kami panjatkan kehadirat Allah SWT, karena berkat rahmat dan karunia-Nya, penulis dapat menyelesaikan laporan penelitian dengan judul “**PERANCANGAN INVERSE KINEMATICS ROBOT MANIPULATOR 4 DEGREE OF FREEDOM**” untuk memenuhi syarat kelulusan pada Departemen Teknik Elektro Otomasi, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember (ITS) Surabaya.

Laporan penelitian ini dapat diselesaikan oleh penulis berkat bantuan, bimbingan, dan dukungan dari berbagai pihak. Penulis ingin mengucapkan terima kasih kepada orang tua dan keluarga tercinta yang selalu memberi dukungan, semangat, dan doa untuk keberhasilan penulis. Terima kasih sebanyak-banyaknya kepada Bapak Slamet Budiprayitno, ST., M.T. selaku dosen pembimbing I dan Ibu Yunafi’atul Aniroh, ST., M.Sc. selaku dosen pembimbing II atas bimbingan dan arahannya dalam proses pengerjaan Tugas Akhir. Tak lupa, terima kasih pula untuk Bapak Imam Arifin, ST., MT., dan Bapak Mohammad Abdul Hady, ST., MT. selaku pembimbing laboratorium yang telah mendidik penulis menjadi lebih baik. Bapak Andri Ashfahani, ST., MT., MSc. selaku dosen wali yang banyak membantu penulis dalam menjalani perkuliahan selama ini serta para dosen Departemen Teknik Elektro Otomasi atas pendidikan dan ilmunya. Keluarga laboratorium otomasi dan sistem komputer serta teman-teman laboratorium AJ-104 atas bantuan dan kerja samanya. Teman-teman dan semua pihak yang tidak dapat penulis sebutkan satu persatu.

Penulis berharap laporan ini dapat bermanfaat bagi pembaca pada umumnya dan penulis pada khususnya. Laporan ini masih jauh dari sempurna, sehingga penulis mengharapkan kritik dan saran dari pembaca yang bersifat membangun.

Surabaya, 19 Juli 2017

Penulis

*Halaman ini sengaja dikosongkan*

# DAFTAR ISI

<b>HALAMAN JUDUL .....</b>	<b>i</b>
<b>PERNYATAAN KEASLIAN TUGAS AKHIR.....</b>	<b>Error! Bookmark not defined.</b>
<b>LEMBAR PENGESAHAN .....</b>	<b>Error! Bookmark not defined.</b>
<b>ABSTRAK .....</b>	<b>ix</b>
<b>ABSTRACT .....</b>	<b>xi</b>
<b>KATA PENGANTAR.....</b>	<b>xiii</b>
<b>DAFTAR ISI .....</b>	<b>xv</b>
<b>DAFTAR GAMBAR .....</b>	<b>xvii</b>
<b>DAFTAR TABEL .....</b>	<b>xix</b>
<b>BAB 1 PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah .....	2
1.4 Tujuan.....	2
1.5 Metodologi.....	2
1.6 Sistematika Penulisan .....	3
<b>BAB 2 KINEMATIKA ROBOT .....</b>	<b>5</b>
2.1 Robot <i>Manipulator</i> .....	5
2.2 Prinsip Dasar Permodelan Matematik Dalam Sistem Robotik .....	5
2.3 Kinematika Robot .....	6
2.3.1 <i>Denavit-Haternberg</i> Parameter.....	6
2.3.2 <i>Homogenous Transformation</i> .....	8
2.3.3 <i>Forward Kinematics</i> .....	8
2.3.4 <i>Inverse Kinematics</i> .....	8
2.4 LabVIEW.....	9
2.5 NI myRIO 1990 .....	9
2.6 Robo Analyzer .....	10
<b>BAB 3 PERENCANAAN SISTEM .....</b>	<b>11</b>
3.1 Blok Fungsional Sistem .....	11
3.2 Konfigurasi Sistem .....	12
3.2.1 Konfigurasi Sistem Robot <i>Manipulator</i> .....	12
3.2.2 Spesifikasi Robot Tetrix .....	13
3.3 Konfigurasi Motor dengan NI myRIO .....	14
3.4 Konfigurasi Persamaan Kinematika.....	15

3.4.1	Geometri Robot <i>Manipulator</i> .....	15
3.4.2	Kaidah <i>Denavit-Haternberg</i> Robot <i>Manipulator</i> .....	17
3.4.3	<i>Homogenous Transformation</i> Robot <i>Manipulator</i> 4 Degree Of Freedom .....	18
3.5	Penyelesaian <i>Forward Kinematics</i> Robot <i>Manipulator</i> 4 Degree Of Freedom .....	20
3.6	Penyelesaian <i>Inverse Kinematics</i> Robot <i>Manipulator</i> 4 Degree Of Freedom .....	20
3.7	Perencanaan <i>Software</i> .....	23
	<b>BAB 4 PENGUJIAN DAN ANALISA DATA .....</b>	<b>27</b>
4.1	Pengujian Robot <i>Manipulator</i> Menggunakan Metode <i>Forward Kinematics</i> .....	27
4.1.1	Perhitungan Manual .....	27
4.1.2	Simulasi RoboAnalyzer .....	29
4.1.3	<i>Real Plant</i> .....	38
4.1.4	Persentase Nilai Kesalahan Berdasarkan Hasil Perhitungan dengan Hasil Pengukuran .....	41
4.2	Pengujian Robot <i>Manipulator</i> Menggunakan Metode <i>Inverse Kinematics</i> .....	42
4.2.1	Perhitungan Manual .....	42
4.2.2	<i>Real Plant</i> .....	45
4.2.3	Persentase Nilai Kesalahan Berdasarkan Hasil Perhitungan dengan Hasil Pengukuran .....	51
	<b>BAB 5 PENUTUP .....</b>	<b>52</b>
	<b>DAFTAR PUSTAKA.....</b>	<b>54</b>
	<b>LAMPIRAN .....</b>	<b>56</b>
	Lampiran 1. <i>Flowchart</i> Keseluruhan Sistem .....	56
	Lampiran 2. Rumus Transformasi Homogen .....	58
	Lampiran 3. <i>Listing Program</i> Metode <i>Forward Kinematics</i> .....	60
	Lampiran 4. <i>Listing Program</i> Metode <i>Inverse Kinematics</i> .....	62
	Lampiran 5. <i>Datasheet</i> NI myRIO .....	64
5.1	<i>Connector Pinouts</i> .....	64
5.2	<i>Digital Input Output</i> NI myRIO .....	65
	<b>RIWAYAT PENULIS .....</b>	<b>66</b>



## DAFTAR GAMBAR

<b>Gambar 2.1</b>	Diagram sistem robotik .....	5
<b>Gambar 2.2</b>	Diagram sistem kontrol robotik berorientasi kinematika.....	6
<b>Gambar 2.3</b>	Cara untuk menentukan parameter $\alpha_i$ dan $\theta_i$ [6].....	7
<b>Gambar 3.1</b>	Konfigurasi NI myRIO dengan motor servo dan motor DC..	11
<b>Gambar 3.2</b>	Konfigurasi NI myRIO dengan 3 motor servo dan 2 motor DC.....	12
<b>Gambar 3.3</b>	Robot <i>Manipulator 4 DOF</i> .....	13
<b>Gambar 3.4</b>	Pengkabelan NI myRIO dengan motor DC dan motor servo	14
<b>Gambar 3.5</b>	Rangkaian pengatur tegangan 5VDC .....	15
<b>Gambar 3.6</b>	Gambar geometri robot <i>manipulator 4 DOF</i> .....	16
<b>Gambar 3.7</b>	Perencanaan <i>Human Machine Interface</i> (HMI).....	24
<b>Gambar 3.8</b>	<i>Diagram block Human Machine Interface</i> pada <i>front panel</i> <i>forward kinematics</i> .....	25
<b>Gambar 3.9</b>	<i>Diagram block</i> rumus <i>forward kinematics</i> .....	25
<b>Gambar 3.10</b>	<i>Diagram block</i> rumus <i>inverse kinematics</i> .....	26
<b>Gambar 3.11</b>	Konversi dari <i>degree</i> ke <i>duty cycle</i> .....	26
<b>Gambar 4.1</b>	Posisi awal robot <i>manipulator</i> pada simulasi RoboAnalyzer	30
<b>Gambar 4.2</b>	Posisi robot <i>manipulator</i> dengan nilai $\theta_1=60^0$ ; $\theta_2=-90^0$ , $\theta_3=30^0$ $\theta_4=90^0$ pada simulasi RoboAnalyzer.....	31
<b>Gambar 4.3</b>	Posisi <i>end effector</i> pada koordinat x .....	31
<b>Gambar 4.4</b>	Posisi <i>end effector</i> pada koordinat y .....	32
<b>Gambar 4.5</b>	Posisi <i>end effector</i> pada koordinat z .....	32
<b>Gambar 4.6</b>	Posisi robot <i>manipulator</i> dengan nilai $\theta_1=90^0$ , $\theta_2=-90^0$ , $\theta_3=0^0$ ; $\theta_4=45^0$ pada simulasi RoboAnalyzer.....	33
<b>Gambar 4.7</b>	Posisi <i>end effector</i> pada koordinat x .....	34
<b>Gambar 4.8</b>	Posisi <i>end effector</i> pada koordinat y .....	34
<b>Gambar 4.9</b>	Posisi <i>end effector</i> pada koordinat z .....	35
<b>Gambar 4.10</b>	Posisi robot <i>manipulator</i> dengan nilai $\theta_1=60^0$ , $\theta_2=-90^0$ , $\theta_3=90^0$ , $\theta_4=45^0$ pada simulasi RoboAnalyzer .....	36
<b>Gambar 4.11</b>	Posisi <i>end effector</i> pada koordinat x .....	36
<b>Gambar 4.12</b>	Posisi <i>end effector</i> pada koordinat y .....	37
<b>Gambar 4.13</b>	Posisi <i>end effector</i> pada koordinat z .....	37
<b>Gambar 4.14</b>	<i>Real experiment</i> menggunakan metode <i>forward kinematics</i> ( <i>case 1</i> ).....	38
<b>Gambar 4.15</b>	<i>Real experiment</i> menggunakan metode <i>forward kinematics</i> ( <i>case 2</i> ).....	39

<b>Gambar 4.16</b>	<i>Real experiment menggunakan metode forward kinematics (case 3).....</i>	41
<b>Gambar 4.17</b>	Posisi <i>end effector</i> pada robot <i>manipulator</i> .....	46
<b>Gambar 4.18</b>	<i>Real experiment menggunakan metode inverse kinematics (case 1).....</i>	46
<b>Gambar 4.19</b>	Posisi <i>end effector</i> pada robot <i>manipulator</i> .....	48
<b>Gambar 4.20</b>	<i>Real experiment menggunakan metode inverse kinematics (case 2).....</i>	49
<b>Gambar 4.21</b>	Posisi <i>end effector</i> pada robot <i>manipulator</i> .....	50
<b>Gambar 4.22</b>	<i>Real experiment menggunakan metode inverse kinematics (case 3).....</i>	50

## DAFTAR TABEL

<b>Tabel 3.1</b>	Tabel spesifikasi robot lengan tetrix .....	14
<b>Tabel 3.2</b>	<i>Denavit-Haternberg</i> parameter robot <i>manipulator</i> .....	18
<b>Tabel 4.1</b>	<i>Denavit-Haternberg</i> parameter robot <i>manipulator</i> pada simulasi RoboAnalyzer .....	29
<b>Tabel 4.2</b>	<i>Denavit-Haternberg</i> parameter pada case 1 .....	30
<b>Tabel 4.3</b>	<i>Denavit-Haternberg</i> parameter pada case 2 .....	33
<b>Tabel 4.4</b>	<i>Denavit-Haternberg</i> parameter pada case 3 .....	35
<b>Tabel 4.5</b>	Pengujian <i>endeffector</i> terhadap perubahan sudut <i>joint</i> menggunakan metode <i>forward kinematics (case 1)</i> .....	39
<b>Tabel 4.6</b>	Pengujian <i>endeffector</i> terhadap perubahan sudut <i>joint</i> menggunakan metode <i>forward kinematics (case 2)</i> .....	40
<b>Tabel 4.7</b>	Pengujian <i>endeffector</i> terhadap perubahan sudut <i>joint</i> menggunakan metode <i>forward kinematics (case 3)</i> .....	41
<b>Tabel 4.8</b>	Persentase nilai kesalahan <i>endeffector</i> terhadap perubahan koordinat x, y, dan z menggunakan metode <i>forward kinematics</i> .....	42
<b>Tabel 4.9</b>	Pengujian <i>endeffector</i> terhadap perubahan sudut <i>joint</i> menggunakan metode <i>inverse kinematics (case 1)</i> .....	47
<b>Tabel 4.10</b>	Pengujian <i>endeffector</i> terhadap perubahan sudut <i>joint</i> menggunakan metode <i>inverse kinematics (case 2)</i> .....	49
<b>Tabel 4.11</b>	Pengujian <i>endeffector</i> terhadap perubahan sudut <i>joint</i> menggunakan metode <i>inverse kinematics (case 3)</i> .....	51
<b>Tabel 4.12</b>	Persentase nilai kesalahan <i>endeffector</i> terhadap perubahan koordinat x, y, dan z menggunakan metode <i>inverse kinematic</i> .....	51

*Halaman ini sengaja dikosongkan*

# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Robot industri adalah komponen utama dalam teknologi otomasi yang dapat berfungsi sebagai layaknya pekerja manusia di dalam pabrik, namun memiliki kemampuan bekerja secara terus menerus tanpa lelah [1]. Salah satu robot yang sering digunakan di industri adalah robot *manipulator*. Hal ini dibuktikan dengan kenaikan penjualan robot industri sebesar 59% yaitu 183.000 unit pada tahun 2016 untuk kawasan Asia saja [2]. Contoh penggunaan robot *manipulator* terdapat pada proses pemindahan barang. Dalam melakukan tugas tersebut, robot *manipulator* memindahkan barang dari satu titik ke titik yang lain. Dalam melakukan perpindahan tersebut diperlukan sebuah metode untuk membuat robot bergerak ke posisi yang diinginkan.

Kebutuhan tersebut membuat banyak penelitian mengenai pergerakan robot *manipulator*, Richard P. Paul telah berhasil menurunkan persamaan kinematika untuk setiap *manipulator* yang ditentukan dari posisi dalam koordinat *Cartesian* dan orientasi dari *end effector* yang diberikan dari koordinat masing-masing *joint* yang ada pada *manipulator* [3]. Kemudian Jin Huang pada tahun 2012 telah membuat simulasi untuk pergerakan *joint* pada *manipulator* menggunakan metode *inverse kinematics* yang lebih efisien dan akurat. [4], namun belum diterapkan pada *manipulator* sesungguhnya. Pada tahun 2013, Yang Si menemukan solusi metode *inverse kinematics* pada robot *manipulator* 4 DOF (Degree of Freedom). Berdasarkan penelitian ini, untuk menyelesaikan permasalahan *inverse kinematics* tergantung dari struktur kinematika dari *manipulator*. Dengan begitu prosedurnya pun berasal dari seberapa panjang perubahan struktur *manipulator*.

Penurunan metode *inverse kinematics* dapat diketahui setelah dilakukan analisa kinematika pada robot *manipulator* tersebut. Model kinematika didapatkan dengan menggambarkan geometri dari robot. Dari gambar geometri tersebut dapat ditemukan *Denavit-Haternberg* Parameter dan matriks transformasi homogenya. Setelah didapatkan hasil permodelan kinematika dari robot *manipulator*, maka dapat dihitung *inverse kinematics* dari robot *manipulator* tersebut. Metode *inverse kinematics* digunakan untuk menghitung sudut-sudut setiap *joint* pada *manipulator* dengan posisi *end effector* yang telah diketahui.

## 1.2 Rumusan Masalah

Secara umum robot *manipulator* pada industri digunakan untuk memindahkan sebuah benda ke titik yang diinginkan sehingga masing-masing *joint* harus bergerak dengan sudut yang sesuai agar dapat mencapai posisi atau titik yang diinginkan.

## 1.3 Batasan Masalah

Robot *manipulator* yang digunakan merupakan robot *manipulator 4 DOF* (*Degree of Freedom*) terdiri dari 4 *joint*, 5 *link*, dan *end effector*. Dimana setiap *joint* pada robot *manipulator* digerakkan menggunakan 2 buah motor DC dan 3 buah motor servo. Perancangan robot *manipulator* difokuskan pada sistem kontrol posisi saja. Parameter yang digunakan dalam analisa kinematika adalah *Denavit-Haternberg Parameter*. Sistem kendali menggunakan sebuah *embedded plant* NI myRIO 1990 serta HMI (*Human Machine Interface*) menggunakan *software* LabVIEW.

## 1.4 Tujuan

*Joint-joint* pada robot *manipulator* dapat bergerak sesuai dengan sudut yg dihasilkan dari hasil rancangan perhitungan *inverse kinematics* dan mendapatkan hasil rancangan perhitungan *inverse kinematics*. Memastikan *endeffector* pada robot *manipulator* dapat bergerak menuju ke posisi yang diinginkan.

## 1.5 Metodologi

Dalam pelaksanaan penelitian perancangan robot *manipulator 4 DOF* (*Degree Of Freedom*) dengan metode *inverse kinematics*, ada beberapa kegiatan yang dapat diuraikan. Diawali dengan tahap persiapan, pada tahap ini dilakukan studi literatur mengenai dasar robot *manipulator* seperti derajat kebebasan, konfigurasi *manipulator*, dan contoh masing-masing jenisnya. Hal ini diperlukan untuk mendukung dalam proses pembelajaran pergerakan robot *manipulator* sebelum memahami konsep dasar *inverse kinematics*.

Pada tahap kedua yaitu tahap perancangan yang dilakukan adalah merancang gambar geometri robot *manipulator*, menentukan *Denavit-Haternberg* parameter, kemudian menyusunnya ke dalam matriks transformasi homogen, dilanjutkan dengan perancangan rangkaian elektronika meliputi pengatur tegangan 5VDC sebagai catu daya untuk motor servo.

Tahap selanjutnya adalah perancangan *hardware* yang dilakukan dengan cara membuat rangkaian konektor dan pengkabelan keseluruhan sistem.

Setelah melakukan perancangan *hardware* maka yang perlu dilakukan selanjutnya adalah tahap pengujian dan analisa. Pengujian dilakukan pada motor servo yang berfungsi sebagai *end effector*, *wrist*, dan *elbow* pada robot *manipulator*. Pada tahap ini pula akan dilakukan pengujian pada metode *forward kinematics* dan *inverse kinematics*.

Pada tahap penyusunan laporan dalam penyusunan laporan diperlukan beberapa hal, pengumpulan seluruh analisa data dari semua percobaan, serta penyusunan referensi.

## **1.6 Sistematika Penulisan**

Pembahasan Tugas Akhir ini akan dibagi menjadi lima Bab dengan sistematika sebagai berikut:

### **BAB I      PENDAHULUAN**

Pada Bab I ini membahas tentang latar belakang, perumusan masalah, batasan masalah, maksud dan tujuan penelitian, sistematika laporan, dan metodologi.

### **BAB II      KINEMATIKA ROBOT**

Bab kinematika robot ini menjelaskan teori dasar di bidang robotika yang dapat dijadikan landasan dan mendukung dalam perencanaan dan pembuatan alat yang dibuat

### **BAB III      PERANCANGAN SISTEM**

Bab III berisi tentang perencanaan dan pembuatan tentang *hardware* yang meliputi desain mekanik dan perancangan *software* yang meliputi program yang akan digunakan untuk menjalankan alat tersebut.

### **BAB IV      PENGUJIAN DAN ANALISIS SISTEM**

Pada Bab IV ini akan membahas pengujian alat dan menganalisa data yang didapat dari pengujian tersebut serta membahas tentang pengukuran, pengujian, dan penganalisaan terhadap alat.

### **BAB V      PENUTUP**

Bab Penutup ini berisi kesimpulan yang didapat dari Tugas Akhir ini dan saran-saran untuk pengembangan alat ini lebih lanjut.

*Halaman ini sengaja dikosongkan*



## BAB 2

### KINEMATIKA ROBOT

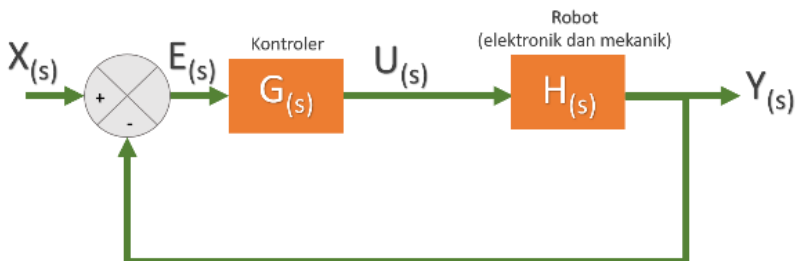
#### 2.1 Robot Manipulator [5]

Robot *manipulator* merupakan jenis robot lengan dengan material lengan pembawa beban. Robot *manipulator* dapat membawa beban dengan lokasi yang berpindah-pindah, berbahan material tipis, lebih ringan, lebih hemat dalam konsumsi daya, hanya memerlukan aktuator yang kecil, lebih mudah dioperasikan, serta lebih murah dalam proses *manufacturing* [6].

Pada robot *manipulator* base-nya tetap dan tak dapat berpindah. Robot lengan memiliki *link* dan *joint*. *Link* pada robot lengan merupakan penghubung antar *joint*. Sedangkan *joint* merupakan penghubung antar *link*. Jumlah dari *joint* suatu *manipulator* biasanya menentukan jumlah *DOF*. *DOF* adalah jumlah arah independen *endeffector* dari sebuah robot yang dapat bergerak.

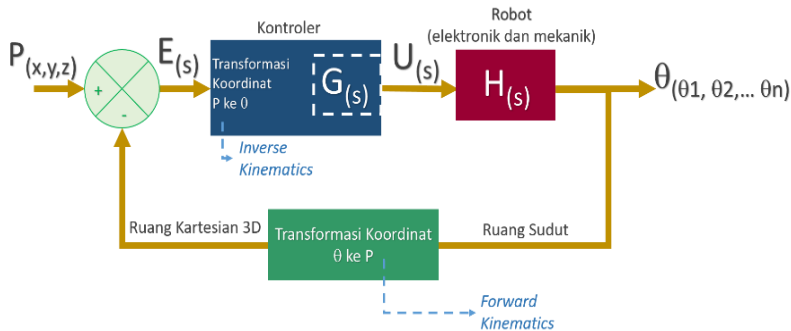
*Joint* robot lengan mirip dengan *joint* dalam tubuh manusia. Namun, setiap *joint* lengan robot hanya memiliki 1 derajat kebebasan sehingga hampir semua kasus, jumlah *joint* yang dimiliki robot lengan sama dengan jumlah *DOF*-nya. Setiap *joint* dihubungkan dengan dua *link*, yaitu *link* masukan dan keluaran.

#### 2.2 Prinsip Dasar Permodelan Matematik Dalam Sistem Robotik [1]



Gambar 2.1 Diagram sistem robotik

Dalam bentuk diagram sistem seperti yang ada pada Gambar 2.1,  $G(s)$  adalah persamaan matematik kontroler.  $H(s)$  adalah persamaan untuk sistem robot secara fisik termasuk aktuator dan sistem elektroniknya. Komponen  $X(s)$  adalah masukan dapat berupa posisi referensi maupun kecepatan referensi. Komponen  $E(s)$  adalah *error* atau kesalahan sedangkan  $U(s)$  adalah keluaran dari kontroler tersebut.  $Y(s)$  merupakan fungsi gerak robot yang diharapkan selalu sama dengan referensi yang didefinisikan pada masukan  $X(s)$ .



**Gambar 2.2** Diagram sistem kontrol robotik berorientasi kinematika

Jika masukan yang digunakan berupa suatu koordinat vector posisi dan orientasi  $P(x,y,z)$  dan keluarannya berupa  $\theta(\theta_1, \theta_2, \dots, \theta_n)$  dimana  $n$  adalah jumlah *DOF*. Maka Gambar 2.1 dapat digambar ulang menjadi Gambar 2.2.

Gambar 2.2 di atas menunjukkan keluaran yang diukur dari gerakan robot adalah domain sudut dari *joint-joint* yang ada. Dalam proses kontrol robot, yang diperlukan oleh *user* untuk melakukan pemrograman atau dalam pemetaan ruang kerja robot (*workspace*) adalah posisi yang dinyatakan sebagai koordinat 3D. Dengan demikian perlu dilakukan transformasi koordinat antara ruang kartesian dengan ruang sudut ini. Di dalam gambar dinyatakan sebagai *inverse kinematics* dan *forward kinematics*. Kombinasi antara  $P$  ke  $\theta$  dengan kontroler  $G(s)$  disebut sebagai kontroler kinematik karena mengandung komponen transformasi ruang kartesian ke ruang sudut. Dengan demikian, keluaran kontroler  $u$  yang bekerja dalam ruang sendi,  $u(\theta_1, \theta_2, \dots, \theta_n)$ .

## 2.3 Kinematika Robot [7]

Kinematika merupakan ilmu yang mempelajari bagaimana gerak dapat terjadi tanpa memperdulikan penyebab terjadinya gerak tersebut.

### 2.3.1 Denavit-Haternberg Parameter [5]

Suatu cara khas representasi analisa hubungan gerak rotasi dan translasi antara lengan-lengan yang terhubung dalam suatu *manipulator* telah diperkenalkan oleh Denavit & Haternberg (1955). Meskipun telah 5 dasawarsa yang lalu, metode ini masih banyak digunakan utamanya untuk pemrograman robot-robot *manipulator* di industri. Mereka memperkenalkan suatu metode yang berguna untuk menetapkan suatu sistem koordinat berorientasi *body*

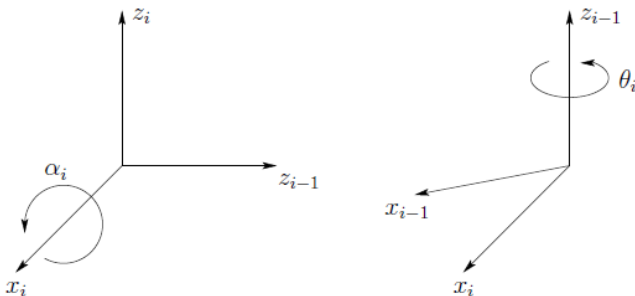
(*body attached frame*) untuk setiap lengan atau *link* yang terhubung dalam suatu struktur hubungan seperti rantai.

Prinsip dasar representasi *Denavit-Hatemberg* parameter adalah melakukan transformasi koordinat antar dua *link* yang berdekatan. Hasilnya adalah suatu matriks (4x4) yang menyatakan sistem koordinat dari suatu *link* dengan *link* yang terhubung pada pangkalnya (*link* sebelumnya). Dalam konfigurasi serial, koordinat (ujung) *link*-1 dihitung berdasarkan *joint*-0 atau *joint* pada tubuh robot. Sistem koordinat *link*-2 dihitung berdasarkan posisi *joint*-1 yang berada di ujung *link*-1 dengan mengasumsikan *link*-1 adalah basis gerakan *link*-2. Demikian seterusnya, *link*-3 dihitung berdasarkan *link*-2, hingga, *link*-ke(*n*) dihitung berdasarkan *link*-ke(*n*-1).

Dengan cara ini maka tiap langkah perhitungan atau transformasi hanya melibatkan sistem 1 *DOF* (*Degree of Freedom*) saja. Terakhir, posisi koordinat tangan atau posisi ujung robot (*end-effector*) akan dapat diketahui. Dalam hal ini, presentasi *Denavit-Hatemberg* parameter menggunakan 4 buah parameter, yaitu  $\theta$ ,  $\alpha$ ,  $d$  dan  $a$  [7] .

Dari persamaan (2.4) diketahui terdapat 4 parameter yaitu  $a_i$  (*link length*),  $\alpha_i$  (*link twist*),  $d_i$  (*link offset*),  $\theta_i$  (*joint angle*). Penamaan tersebut diperoleh dari beberapa aspek spesifik dari hubungan geometri antar dua koordinat *frame*. Persamaan (2.4) dapat digunakan untuk merepresentasikan matriks transformasi homogen.

Parameter  $a$  adalah jarak antar sumbu  $z_0$  dan  $z_1$  dan diukur sepanjang sumbu  $x_1$ . Parameter  $\alpha$  adalah sudut antara sumbu  $z_0$  dan  $z_1$ , diukur dari garis normal ke sumbu  $x_1$  (Gambar 2.3). Parameter  $d$  adalah jarak antara *origin*  $o_0$  dann perpotongan sumbu  $x_1$  dengan  $z_0$ , diukur sepanjang sumbu  $z_0$ . Sedangkan  $\theta$  adalah sudut antar sumbu  $x_0$  dan  $x_1$  diukur dari  $z_0$  (Gambar 2.3).



**Gambar 2.3** Cara untuk menentukan parameter  $\alpha_i$  dan  $\theta_i$  [6]

### 2.3.2 Homogenous Transformation [5]

Matriks transformasi homogen adalah sebuah matriks 4x4 yang didefinisikan untuk memetakan vektor *homogenous position* dari satu koordinat ke koordinat yang lain [8]. Matriks tersebut dapat dituliskan sebagai berikut:

$${}^A T_B = \begin{bmatrix} {}^A R_{B(3 \times 3)} & {}^A q_{(3 \times 1)} \\ 0_{(1 \times 3)} & 1_{(1 \times 1)} \end{bmatrix} \quad (2.1)$$

$${}^{i-1} A_i = Rot_{z,\theta} Trans_{z,d} Rot_{x,\alpha} Trans_{x,a} \quad (2.2)$$

$${}^{i-1} A_i = \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & 0 \\ \sin \theta_i & \cos \theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha_i & -\sin \alpha_i & 0 \\ 0 & \sin \alpha_i & \cos \alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.3)$$

$${}^{i-1} A_i = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

### 2.3.3 Forward Kinematics [5]

Masalah *forward kinematics* berkaitan dengan hubungan antara *joint* robot *manipulator* dan posisi serta orientasi *end effector*. Dinyatakan lebih formal, masalah *forward kinematics* adalah menentukan posisi dan orientasi *end effector*, mengingat nilai untuk variabel gabungan robot.

### 2.3.4 Inverse Kinematics [5]

*Inverse kinematics* mengacu pada penggunaan persamaan kinematika robot untuk menentukan parameter bersama yang memberikan posisi yang diinginkan pada posisi akhir atau *end effector*. *Inverse kinematics* mengubah rencana gerak menjadi nilai yang harus diberikan bagi aktuator atau penggerak dalam pergerakan robot. *Inverse kinematics* adalah metode untuk menghitung nilai-nilai rotasi bersama derajat kebebasan individu melalui rotasi dan posisi yang telah ditetapkan. *Inverse kinematics* adalah teknik yang digunakan dalam model animasi kompleks dan *rig* gerak dengan sejumlah *joint*. Perbedaan mendasar dengan *forward kinematics*, *inverse kinematics* ditentukan dari

pergerakan *link* berdasarkan sudut akhir dari beberapa *joint* yang mendefinisikan gerakan.

## 2.4 LabVIEW

LabVIEW adalah perangkat lunak sistem rekayasa untuk aplikasi yang memerlukan pengujian, pengukuran, dan kontrol dengan akses cepat ke wawasan perangkat keras dan data. LabVIEW (singkatan dari Laboratory Virtual Instrumentation Engineering Workbench) adalah perangkat lunak komputer untuk pemrosesan dan visualisasi data dalam bidang akuisisi data, kendali instrumentasi serta automasi industri yang pertama kali dikembangkan oleh perusahaan National Instruments pada tahun 1986.

## 2.5 NI myRIO 1990 [8]

myRIO adalah sebuah perangkat keras dimana pengguna dapat memanipulasi fungsi – fungsinya untuk membuat berbagai sistem. myRio menggunakan ARM prosesor yang juga sebuah FPGA prosesor. Dengan menggunakan fitur-fitur yang ada dapat dibuat sistem yang lebih kompleks. myRIO menggunakan LabVIEW sebagai IDE. LabVIEW digunakan untuk membuat aplikasi yang akan menjalankan perangkat keras atau fitur – fitur yang ada pada myRIO. Pada *board* myRIO terdapat beberapa *peripheral* yang umum yang dapat digunakan seperti *bluetooth*, *wifi*, LED, *accelerometer*, *push button*, *analog input* dan *output*, RAM. Dapat juga dihubungkan ke sebuah *integrated circuit* atau IC (*Integrated Circuit*) untuk membuat sistem yang lebih kompleks. Atau dihubungkan ke *board* lain yang memang diperlukan dalam membuat sebuah sistem.

Sebuah *embedded processor* adalah mikroprosesor yang digunakan dalam sistem *embedded*. Prosesor ini biasanya memiliki ukuran lebih kecil dan mengkonsumsi daya yang lebih kecil. *Embedded processor* dapat dibagi menjadi dua kategori: mikroprosesor biasa dan mikrokontroler. Mikrokontroler memiliki *peripheral* yang lebih banyak pada chipnya. Pada dasarnya, *embedded processor* adalah *chip* CPU yang digunakan dalam sistem yang bukan merupakan *general purpose workstation* seperti laptop atau desktop. *Embedded processor* hanyalah sebuah perangkat komputasi yang ditempatkan di dalam sistem yang dikontrolnya. Sebuah *embedded processor* yang ada pada sistem menangani semua perhitungan dan operasi logis dari komputer. *Embedded processor* juga menangani tugas-tugas seperti menyimpan dan mengambil data dari memori, dan pengolahan data dari *input* atau *output*. *Embedded processor* sering bekerja sebagai bagian dari sistem komputer, bersama memori dan perangkat *input* dan *output*.

## 2.6 Robo Analyzer

RoboAnalyzer adalah perangkat lunak berbasis model 3D yang bisa digunakan untuk mengajarkan dan mempelajari konsep Robotika. Ini adalah produk yang berkembang yang dikembangkan di Laboratorium Mekatronika, Departemen Teknik Mesin di IIT Delhi, New Delhi, India, di bawah bimbingan Prof. S.K. Saha dalam rangka mendukung pembelajaran atau pengajaran dari topik yang dibahas dalam bukunya "*Introduction to Robotics*" yang diterbitkan oleh Tata McGraw Hill, New Delhi (Edisi Pertama tahun 2008 dan Edisi Kedua tahun 2014) dan juga dibawa keluar di Meksiko (2009 di Spanyol ), China (2010) dan Singapura (2010).

Virtual Robot Module, bagian dari RoboAnalyzer, telah dikembangkan sebagai aplikasi yang memiliki gerak sendi dan *cartesian*. Ini juga telah dibuat sebagai server COM, dengan menggunakan mana seseorang dapat mengintegrasikan VRM dengan MATLAB, MS Excel dan aplikasi lain yang memiliki antarmuka COM. Ini juga telah terintegrasi dengan Robotics Toolbox untuk MATLAB oleh Prof. Peter Corke dari QUT, Australia. Keterangan lebih lanjut disini.

Kami juga menyiapkan seperangkat Percobaan Virtual menggunakan RoboAnalyzer, yang dapat digunakan untuk melengkapi kursus robotika. Keterangan lebih lanjut disini. Mirip dengan RoboAnalyzer, perangkat lunak lain yang dikembangkan oleh kita untuk mengajar dan belajar Mekanisme diberi nama MechAnalyzer. Rincian lebih lanjut tentang MechAnalyzer.

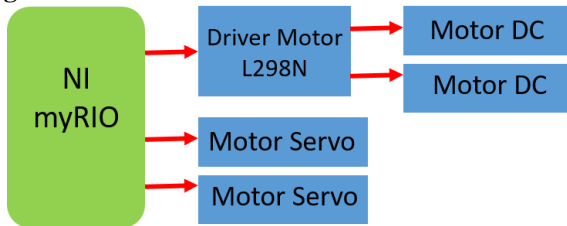
Matematika yang terlibat dalam studi robotika, misalnya, *forward kinematics* dan *inverse kinematics* pada awalnya sulit dipahami oleh siswa dan hal yang sama terjadi pada seorang guru untuk menyampaikan esensi matematika robotika kepada siswa. Hal ini disebabkan oleh fakta bahwa, misalnya, forward melibatkan transformasi 3D dan lain-lain. Perlu dicatat juga bahwa robot industri diwakili dengan menggunakan parameter Denavit dan Haternberg (DH) yang sulit untuk dipahami dan divisualisasikan dalam 3D.

## BAB 3

### PERENCANAAN SISTEM

Pada Tugas Akhir sebelumnya [9], telah dibahas mengenai robot *manipulator* namun tidak menggunakan metode untuk menghitung pergerakan robot *manipulator*. Apabila robot *manipulator* diharapkan dapat bergerak ke posisi yang diinginkan maka diperlukan sebuah metode untuk membantu pergerakannya. Perencanaan sistem pada penelitian ini fokus pada pergerakan robot *manipulator* menggunakan analisa kinematika. Gambar geometri, transformasi homogen, hingga *Denavit-Haternberg* parameter akan dibahas pada bab ini. Perhitungan rumus *forward kinematics* didapatkan dari hasil matriks transformasi homogen *endeffector* terhadap *base* sedangkan perhitungan rumus *inverse kinematics* didapatkan berdasarkan pendekatan aljabar.

#### 3.1 Blok Fungsional Sistem



**Gambar 3.1** Konfigurasi NI myRIO dengan motor servo dan motor DC

Robot *manipulator* tersebut memiliki 2 buah motor DC, dimana satu motor DC berfungsi sebagai *base* dan satu lagi sebagai *shoulder*, serta memiliki 3 buah motor servo sebagai *elbow*, *wrist*, dan *end effector*. Pada Gambar 3.1 motor servo dan *deiver* motor yang digunakan untuk mendrive motor DC akan dikoneksikan dengan NI myRIO.

Masukan sistem robot *manipulator* adalah posisi. Dimana posisi berada di dalam koordinat 3 dimensi (x, y, dan z), karena pergerakan masing-masing *joint* pada robot *manipulator* bergerak sesuai derajat yang diinginkan maka diperlukan untuk membangun hubungan untuk mengkonversi koordinat ke dalam derajat (*degree*). [10] Dalam hal ini digunakannya metode *inverse kinematics*.

Hasil konversi akan menghasilkan derajat untuk mengatur posisi motor servo. Masukan berupa sudut menjadi *setpoint* pada kontrol posisi motor DC.

Dimana dari derajat (*degree*) dikonversi dalam bentuk pulsa, nilai kesalahan (*error*) dihasilkan dari hasil selisih nilai *counter* yang terbaca oleh sensor *rotary encoder*. Sedangkan sinyal *error* tersebut diolah oleh kontroler agar dapat dikonversi sesuai dengan nilai PWM (*Pulse Width Modulation*), sehingga motor DC dapat bekerja pada posisi yang diinginkan.

### 3.2 Konfigurasi Sistem

Konfigurasi sistem berikut ini membahas mengenai keseluruhan sistem yang akan digunakan. Pada subbab ini berisi tentang dengan konfigurasi sistem robot manipulator dan spesifikasi robot manipulator yang digunakan.

#### 3.2.1 Konfigurasi Sistem Robot *Manipulator*

Robot lengan dengan menggunakan motor servo difungsikan sebagai *elbow*, *wrist*, *end effector* menggunakan motor servo tipe HITEC 485HB serta motor DC berfungsi sebagai *base* dan *shoulder*, yang akan dikoneksikan dengan menggunakan *embedded controller* NI myRIO. Pengkabelan motor servi dan motor DC dengan NI myRIO seperti pada Gambar 3.2



**Gambar 3.2** Konfigurasi NI myRIO dengan 3 motor servo dan 2 motor DC

Adapun perencanaan konfigurasi sistem robot lengan meliputi 3 tahap, yakni:

1. Perencanaan modifikasi robot lengan  
Modifikasi pada robot lengan agar mampu memenuhi gerakan 4 *Joint* (4 *DOF*) pada robot lengan, yaitu *Base*, *Shoulder*, *Elbow*, *Wrist*, dan *End Effector*.
2. Perencanaan metode *inverse kinematics*



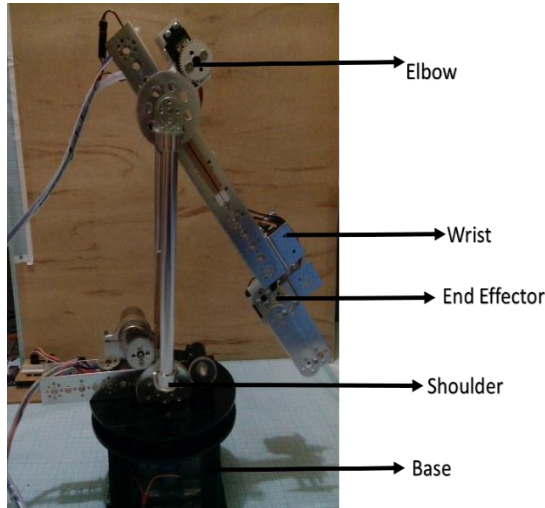
Perencanaan metode digunakan untuk memenuhi kebutuhan robot *manipulator* yang harus menuju titik ke arah yang diinginkan. Penggunaan metode *inverse kinematics* diharapkan agar *joint* yang ada pada robot *manipulator* mampu bergerak seefektif dan seefisien mungkin.

3. Perencanaan *Software*

Perencanaan *software* meliputi Program HMI (*Human-Machine Interface*) dengan menggunakan *software* LabVIEW 2015.

### 3.2.2 Spesifikasi Robot Tetrix

Robot Lengan Tetrix memiliki empat derajat kebebasan (4 *DOF*) yaitu *Base*, *Shoulder*, *Elbow*, dan *Wrist* serta dengan satu buah *End of Effector* seperti pada Gambar 3.3. Tabel 3.1 merupakan tabel spesifikasi robot *manipulator* 4 *DOF*.



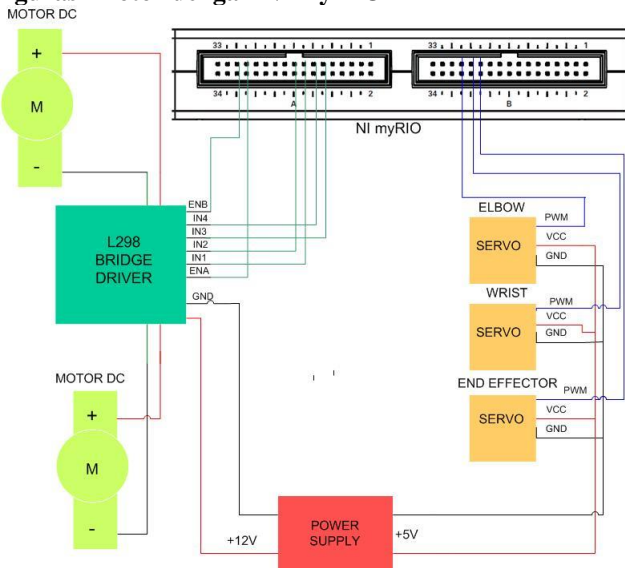
**Gambar 3.3** Robot Manipulator 4 *DOF*

Pada Tabel 3.1 berisi tentang spesifikasi dari robot manipulator yang digunakan. Robot manipulator yang digunakan memiliki 4 derajat kebebasan dengan tipe rotasi. Konstruksi robot manipulator memiliki keterbatasan, untuk *joint base* sejauh  $-360^0$  sampai  $360^0$ . *Joint shoulder* sejauh  $0^0$  sampai  $90^0$ . *Joint elbow* sejauh  $-90^0$  sampai  $90^0$ . Sedangkan *joint wrist* sejauh  $-90^0$  sampai  $90^0$ .

**Tabel 3.1** Tabel spesifikasi robot lengan tetrix

MODEL		ROBOT LENGAN TETRIX
Koordinat Robot		4 derajat kebebasan (4 <i>DOF</i> ) + <i>End Effector</i>
Batasan Gerak	<i>Base</i>	-360° s/d 360°
	<i>Shoulder</i>	0° s/d 90°
	<i>Elbow</i>	-90° s/d 90°
	<i>Wrist</i>	-90° s/d 90°
Material		Alumunium

### 3.3 Konfigurasi Motor dengan NI myRIO



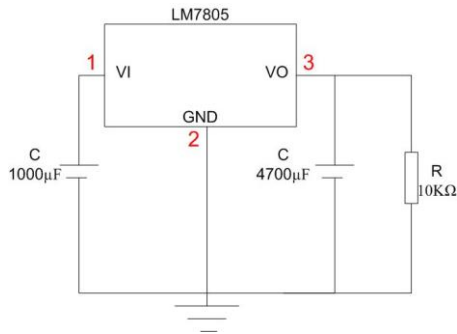
**Gambar 3.4** Pengkabelan NI myRIO dengan motor DC dan motor servo

Gambar 3.4 merupakan pengkabelan antara NI myRIO, motor DC beserta driver motor tipe L298N, serta motor servo. Motor DC yang digunakan berjumlah 2 buah berfungsi sebagai *base* dan *shoulder* pada robot *manipulator*. Sedangkan 3 motor servo berfungsi sebagai *elbow*, *wrist*, dan *end effector* pada robot *manipulator*.

Motor DC yang digunakan adalah motor DC *gearbox* 12VDC 2A. Sedangkan NI myRIO memiliki tegangan keluaran 5V dengan arus maksimal

sebesar 100mA. Maka dari itu diperlukan sebuah *power supply* sebagai sumber tegangan untuk menggerakkan motor DC.

Motor servo yang digunakan membutuhkan tegangan masukan sebesar 5VDC. Walaupun dapat digunakan sumber tegangan menggunakan NI myRIO saja. Namun terbatasnya pin VCC (5VDC) pada myRIO sehingga sumber tegangan motor servo menggunakan *power supply* yang diberi regulator tegangan 5VDC.



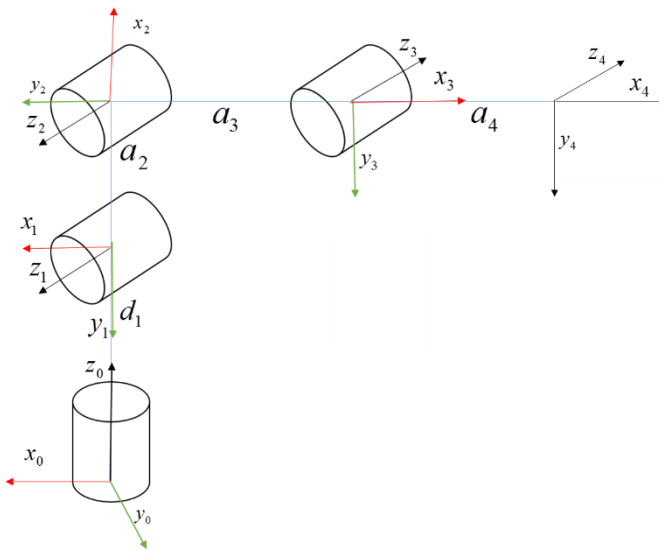
**Gambar 3.5** Rangkaian pengatur tegangan 5VDC

### 3.4 Konfigurasi Persamaan Kinematika

Dalam menentukan persamaan kinematika pada robot *manipulator* diperlukan beberapa hal yaitu yang pertama menentukan gambar geometri dari robot *manipulator*. Kemudian menentukan sumbu x, y, dan z pada masing-masing *joint* yang ada pada robot *manipulator*. Setelah itu membangun transformasi homogen, dilanjutkan dengan menentukan *Denavit-Haternberg* Parameter untuk mendapatkan rumus *forward kinematics* sehingga dapat dilanjutkan dalam penyelesaian *inverse kinematics*.

#### 3.4.1 Geometri Robot Manipulator

Pendeskripsian robot *manipulator* diawali dengan perancangan karakteristik geometri gerak robot (posisi) tanpa dipegaruhi gaya pada *joint-joint* yang terdapat di robot *manipulator* tersebut. Sebelum melakukan analisa kinematika pada robot *manipulator* maka perlu dibangun sebuah geometri robot *manipulator*, sesuai posisi dan orientasi masing-masing *joint*.



**Gambar 3.6** Gambar geometri robot *manipulator* 4 DOF

Seperti yang terlihat pada Gambar 3.6, robot *manipulator* memiliki 3 *joint*, *joint* 1 adalah *base*, *joint* 2 adalah *elbow*, *joint* 3 adalah *wrist*.

Untuk mendapatkan gambar geometri diawali dengan menentukan  $J_i$  hingga letak  $J_{i+1}$  setelah itu menemukan  $L_{i-1}$  hingga  $L_i$ . Sumbu Z ditentukan sesuai arah putar dari *joint*. Setelah sumbu Z didapatkan maka dilanjutkan untuk mencari koordinat X. Dimana letak koordinat  $X_i$  adalah sepanjang garis normal (*common normal*) diantara  $Z_{i-1}$  dan  $Z_i$  (jika ada). Namun apabila  $Z_{i-1}$  berpotongan terhadap  $Z_i$  maka arah  $X_i$  adalah sesuai dengan arah sumbu X pada *joint* sebelumnya.

Setelah selesai menentukan arah  $X_i$  dan  $Z_i$ , letak  $Y_i$  dapat diketahui berdasarkan kaidah tangan kanan. Arah  $O_n$  pada *end effector* searah dengan  $O_{n-1}$ . Sedangkan letak  $X_n$  berpotongan tegak lurus terhadap  $Z_{n-1}$ . Apabila *joint* pada *end effector* merupakan rotasi maka, arah  $Z_n$  parallel dengan  $Z_{n-1}$ . Namun pada *end effector* dengan *joint* tipe prismatic, maka arah  $Z_n$  bebas. Arah  $Y_n$  sesuai dengan kaidah tangan kanan. [11]

### 3.4.2 Kaidah Denavit-Haternberg Robot Manipulator

*Denavit-Haternberg* Parameter merupakan sebuah metode untuk analisa kinematika dari robot *manipulator*. Untuk mendapatkan nilai-nilai *Denavit- Haternberg* Parameter dapat dilakukan dengan cara berikut [7] :

- Langkah 1 : Cari dan beri label sumbu gabungan  $Z_0, \dots, Z_{n-1}$
- Langkah 2 : Tetapkan bingkai dasar. Tetapkan asal mana saja di sumbu  $z_0$ . Sumbu  $x_0$  dan  $y_0$  dipilih dengan mudah untuk membentuk bingkai tangan kanan. Untuk  $i = 1$  sampai dengan  $n-1$ , lakukan langkah 4 sampai 6.
- Langkah 3 : Temukan asal letak *origin*  $O_i$  dimana normal normal untuk  $z_i$  dan  $z_{i-1}$ . Memotong  $z_i$  Jika  $z_i$  berpotongan  $z_{i-1}$  cari  $O_i$  di persimpangan ini. Jika  $z_i$  dan  $z_{i-1}$  sejajar, cari  $O_i$  dalam posisi nyaman sepanjang  $z_i$ .
- Langkah 4 : Tetapkan  $x_i$  di sepanjang garis normal (*common normal*) antara  $z_{i-1}$  dan  $z_i$ ,  $O_i$ , atau ke arah garis normal (*common normal*) ke bidang  $z_{i-1}$  ke  $z_i$  jika  $z_{i-1}$  dan  $z_i$  memotong.
- Langkah 5 : Tetapkan  $y_i$  untuk melengkapi bingkai tangan kanan
- Langkah 6 : Tetapkan bingkai *end-effector*  $o_n x_n y_n z_n$ . Dengan asumsi *joint* ke- $n$ , atur  $z_n$  sama dengan sepanjang arah  $z_{n-1}$ . Tetapkan  $y_n = s$  ke arah penutupan *gripper* dan atur  $x_n = n$  sebagai  $s \times a$ .
- Langkah 7 : Buat tabel parameter link  $a_i, d_i, \alpha_i, \theta_i$ .
  - $a_i$  = jarak sepanjang  $x_i$  dari  $O_i$  ke persimpangan  $x_i$  dan sumbu  $z_i$ .
  - $d_i$  = jarak sepanjang  $z_{i-1}$  dari  $O_{i-1}$  ke persimpangan  $x_i$  dan sumbu  $z_{i-1}$ .  $d_i$  variabel jika *joint* prismatik.
  - $\alpha_i$  = sudut antara  $z_{i-1}$  dan  $z_i$  diukur sekitar  $x_i$ .
  - $\theta_i$  = sudut antara  $x_{i-1}$  dan  $x_i$  diukur sekitar  $z_{i-1}$ .  $\theta_i$  adalah variabel jika *joint* berupa rotasi (*revolute*).
- Langkah 8 : Bentuk matriks transformasi homogen  ${}^{i-1}A_i$  dengan cara mengganti parameter di atas menjadi (2.4).

Setelah menentukan parameter yang akan digunakan maka langkah selanjutnya adalah menyusunnya dalam sebuah tabel sebagai berikut.

**Tabel 3.2** *Denavit-Haternberg* parameter robot *manipulator*

<i>Link</i>	$d_i$	$a_i$	$\alpha_i$	$\theta_i$
<b>1</b>	$d_1$	0	-90	$\theta_1^*$
<b>2</b>	0	$a_2$	0	$\theta_2^*$
<b>3</b>	0	$a_3$	180	$\theta_3^*$
<b>4</b>	0	$a_4$	0	$\theta_4^*$

Pada Tabel 3.2 merupakan tabel *Denavit-Haternberg* Parameter robot *manipulator* 4 *DOF*. Karena pada *link* 1 memiliki *offset* maka *link* 1 memiliki nilai  $d_1$ . *Joint* 1 dan *joint* 2, arah koordinat  $z_0$  dan  $z_1$  tegak lurus maka pada *link* 1 tidak memiliki nilai  $a$ . Perubahan sudut dari sumbu  $z_0$  menuju ke  $z_1$  terhadap sumbu  $x_1$  adalah  $-90^0$  (arah berkebalikan dari arah jarum jam). *Joint* 1 hingga *joint* 4 adalah revolute sehingga nilai  $\theta$  adalah variabel. *Link* 2 tidak memiliki *offset* sehingga nilai parameter  $d$ -nya adalah 0. Pada *joint* 2 dan *joint* 3, arah sumbu  $z_1$  dan  $z_2$  adalah paralel sehingga *link* 2 memiliki nilai  $a_2$ . Tidak adanya perubahan sudut dari sumbu  $z_1$  menuju ke  $z_2$  terhadap sumbu  $x_2$  menyebabkan nilai parameter  $\alpha$  adalah 0. Untuk *link* 3 tidak memiliki *offset* juga terhadap *joint* 4, sehingga nilai  $d_3$  adalah 0 namun arah sumbu  $z_2$  dan  $z_3$  paralel menyebabkan *link* 3 memiliki nilai  $a_3$ . Perubahan sudut dari sumbu  $z_2$  menuju ke sumbu  $z_3$  terhadap sumbu  $x_3$  adalah  $180^0$  jika ditinjau dari Gambar 3.6. *Link* 4 menghubungkan *joint* 4 dan *end effector*. *Joint* 4 dan *end effector* tidak memiliki *offset*, namun arah  $z_3$  dan  $z_4$  adalah paralel sehingga memiliki nilai parameter  $a$ . Namun, nilai  $\alpha_4$  adalah 0 dikarenakan tidak ada perubahan sudut antara sumbu  $z_3$  ke sumbu  $z_4$  terhadap  $x_4$ . Berdasarkan *real plant* pada robot *manipulator*, nilai  $d_1$  adalah 9 cm, nilai  $a_2$  adalah 23 cm, nilai  $a_3$  adalah 18cm, dan nilai  $a_4$  adalah 6.5cm

### 3.4.3 Homogenous Transformation Robot Manipulator 4 Degree Of Freedom

$${}^A p = {}^A R_B {}^B p + {}^A q \quad (3.1)$$

Persamaan (3.1) mendeskripsikan posisi dari titik pada *rigid body* terhadap posisi dari origin 1 dan orientasi terhadap *moving frame* B [12]

Dalam penyelesaian permasalahan kinematika pada robot *manipulator* maka diperlukan menggambarkan hubungan dari *frame* referensi ke masing-

masing *links* pada robot *manipulator*. Selanjutnya melakukan perhitungan dari setiap pergerakan masing-masing *links* yang disebabkan oleh perpindahan *joints*.

Maka dari itu diperlukan beberapa konvensi untuk mendeskripsikan posisi dan orientasi dari robot *manipulator* dan pergerakannya di dalam ruang. Pada bentuk geometri robot *manipulator* pada Gambar 3.6 maka pada masing-masing *joints* menjadi :

Dimana  ${}^{i-1}A_i$  merepresentasikan hubungan antara *joint* ke  $i-1$  dan *joint* ke  $i$ . Dari persamaan maka dapat dibuat hubungan antar *joint* menjadi seperti di bawah ini :

*Joint 1 terhadap joint 0 :*

$${}^0A_1 = \begin{bmatrix} c_1 & 0 & -s_1 & 0 \\ s_1 & 0 & c_1 & 0 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.2)$$

*Joint 2 terhadap joint 1 :*

$${}^1A_2 = \begin{bmatrix} c_2 & -s_2 & 0 & a_2c_2 \\ s_2 & c_2 & 0 & a_2s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.3)$$

*Joint 3 terhadap joint 2 :*

$${}^2A_3 = \begin{bmatrix} c_3 & s_3 & 0 & a_3c_3 \\ s_3 & -c_3 & 0 & a_3s_3 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.4)$$

*Joint 4 terhadap joint 3 :*

$${}^3A_4 = \begin{bmatrix} c_4 & -s_4 & 0 & a_4c_4 \\ s_4 & c_4 & 0 & a_4s_4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.5)$$

$${}^0A_4 = {}^0A_1 {}^1A_2 {}^2A_3 {}^3A_4 \quad (3.6)$$

$${}^0A_4 = \begin{bmatrix} {}^0R_4 & {}^4P \\ 0_{(1 \times 3)} & 1_{(1 \times 1)} \end{bmatrix}$$

$$\begin{aligned}
{}^0R_4 &= \begin{bmatrix} c_4c_1c_{23} + s_4c_1s_{23} & c_4c_1s_{23} - s_4c_1c_{23} & s_1 \\ c_4s_1c_{23} + s_4s_1s_{23} & c_4s_1s_{23} - s_4s_1c_{23} & -c_1 \\ s_4c_{23} - c_4s_{23} & c_4c_{23} + s_4s_{23} & 0 \end{bmatrix} \quad (3.7) \\
{}^4p &= \begin{bmatrix} a_2c_1c_2 - a_4c_4c_1c_{23} + a_4s_4c_1s_{23} + a_3c_1c_{23} \\ a_2s_1c_2 - a_4c_4s_1c_{23} + a_4s_4s_1s_{23} + a_3s_1c_{23} \\ d_1 - a_2s_2 - a_3s_{23} - a_4c_4s_{23} + a_4s_4c_{23} \end{bmatrix}
\end{aligned}$$

Persamaan (3.2) merupakan matriks transformasi homogen pada *link* 0 dan *link* 1 begitu pula dengan persamaan (3.3) yang merepresentasikan hubungan antara *link* 1 dan *link* 2. Hubungan antara *link* 2 dan *link* 3 direpresentasikan dengan matriks homogen pada persamaan (3.4). *Link* 3 dan *link* 4 pun dituliskan dengan persamaan (3.5). Setelah hubungan antar *joint* diketahui, maka perlu mencari hubungan dari *end effector* terhadap *base*.

### 3.5 Penyelesaian *Forward Kinematics Robot Manipulator 4 Degree Of Freedom*

Dari persamaan (3.7) maka didapatkan rumus untuk *forward kinematics* menjadi sebagai berikut :

$$\begin{bmatrix} px \\ py \\ pz \\ 1 \end{bmatrix} = \begin{bmatrix} a_2c_1c_2 + a_4c_4c_1c_{23} + a_4s_4c_1s_{23} + a_3c_1c_{23} \\ a_2s_1c_2 + a_4c_4s_1c_{23} + a_4s_4s_1s_{23} + a_3s_1c_{23} \\ d_1 - a_2s_2 - a_3s_{23} - a_4c_4s_{23} + a_4s_4c_{23} \\ 1 \end{bmatrix} \quad (3.8)$$

Setelah mendapatkan rumus *forward kinematics*, maka dapat ditentukan posisi akhir dari *end effector*.

### 3.6 Penyelesaian *Inverse Kinematics Robot Manipulator 4 Degree Of Freedom*

Robot *manipulator* pada *real plant* memiliki permasalahan pada aktuator di *joint* shoulder sehingga diasumsikan nilai  $\theta_2$  dibuat konstan bernilai -90, sesuai letak geometrinya.

$$({}^0A_1)^{-1}({}^0T_4) = {}^1A_2 {}^2A_3 {}^3A_4 \quad (3.9)$$



$$\begin{bmatrix} \frac{c_1}{c_1^2+s_1^2} & \frac{s_1}{c_1^2+s_1^2} & 0 & 0 \\ 0 & 0 & 1 & -d_1 \\ \frac{s_1}{c_1^2+s_1^2} & \frac{-c_1}{c_1^2+s_1^2} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.10)$$

$$= \begin{bmatrix} c_4c_{23} + s_4s_{23} & c_4s_{23} - s_4c_{23} & 0 & a_2c_2 + a_3c_{23} + a_4c_4c_{23} + a_4s_4s_{23} \\ c_4c_{23} - s_4s_{23} & -c_4c_{23} - s_4s_{23} & 0 & a_2s_2 + a_3s_{23} + a_4c_4s_{23} - a_4s_4c_{23} \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} c_1n_x + s_1n_y & c_1s_x + s_1s_y & c_1a_x + s_1a_y & c_1p_x + s_1p_y \\ n_z & s_z & a_z & d_1 - p_z \\ s_1n_x - c_1n_y & s_1s_x - c_1s_y & s_1a_x - c_1a_y & s_1p_x - c_1p_y \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.11)$$

$$= \begin{bmatrix} c_4c_{23} + s_4s_{23} & c_4s_{23} - s_4c_{23} & 0 & a_2c_2 + a_3c_{23} + a_4c_4c_{23} + a_4s_4s_{23} \\ c_4c_{23} - s_4s_{23} & -c_4c_{23} - s_4s_{23} & 0 & a_2s_2 + a_3s_{23} + a_4c_4s_{23} - a_4s_4c_{23} \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Nilai  $\theta_2$  bernilai  $-90^\circ$ , dari persamaan trigonometri pada persamaan (3.17) sehingga didapatkan persamaan menjadi persamaan (3.18)

$$\cos(a - 90^\circ) = \sin(a) \quad (3.12)$$

$$\sin(a - 90^\circ) = -\cos(a)$$

$$\begin{bmatrix} c_1n_x + s_1n_y & c_1s_x + s_1s_y & c_1a_x + s_1a_y & c_1p_x + s_1p_y \\ n_z & s_z & a_z & d_1 - p_z \\ s_1n_x - c_1n_y & s_1s_x - c_1s_y & s_1a_x - c_1a_y & s_1p_x - c_1p_y \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.13)$$

$$= \begin{bmatrix} c_4s_3 - s_4c_3 & -c_4c_3 - s_4s_3 & 0 & a_3s_3 + a_4c_4s_3 - a_4s_4c_3 \\ c_4s_3 + s_4c_3 & -c_4s_3 + s_4c_3 & 0 & -a_2 - a_3c_3 - a_4c_4c_3 - a_4s_4s_3 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$c_1p_x + s_1p_y = a_3s_3 + a_4c_4s_3 - a_4s_4c_3 \quad (3.14)$$

$$d_1 - p_z = -a_2 - a_3 c_3 - a_4 c_4 c_3 - a_4 s_4 s_3 \quad (3.15)$$

$$-s_1 p_x + c_1 p_y = 0 \quad (3.16)$$

Dari persamaan (3.19) maka didapatkan :

$$\frac{s_1}{c_1} = \frac{p_y}{p_x} \quad (3.17)$$

$$\tan \theta_1 = \frac{p_y}{p_x} \quad (3.18)$$

$$\theta_1 = A \tan \left( \frac{p_y}{p_x} \right) \quad (3.19)$$

$$\theta_1 = A \tan(p_x, p_y) \quad (3.20)$$

Dibuat permasalahan sebagai berikut untuk mempermudah mencari hukum cosinus sebagai berikut :

$$\phi = \theta_4 - \theta_3 \quad (3.21)$$

$$k_1 = a_3 s_3 + a_4 s_\phi \quad (3.22)$$

$$k_2 = -a_3 c_3 - a_4 s_\phi \quad (3.23)$$

Dimana :

$$k_1 = c_1 p_x + s_1 p_y \quad (3.24)$$

$$k_2 = d_1 - p_z + a_2 \quad (3.25)$$

Dari persamaan (3.25) dan (3.26) kemudian dijumlahkan hasil kuadrat di atas menjadi sebagai berikut :

$$k_1^2 + k_2^2 = a_3^2 + a_4^2 + 2a_3 a_4 (c_3 c_\phi + s_3 s_\phi) \quad (3.26)$$

$$\frac{k_1^2 + k_2^2 - a_3^2 - a_4^2}{2a_3 a_4} = s_3 (c_4 s_3 - s_4 c_3) + c_3 (c_4 c_3 + s_4 s_3) \quad (3.27)$$

$$\frac{k_1^2 + k_2^2 - a_3^2 - a_4^2}{2a_3 a_4} = s_3^2 c_4 - s_4 c_3 s_3 + c_3^2 c_4 + s_4 s_3 c_3 \quad (3.28)$$

$$\frac{k_1^2 + k_2^2 - a_3^2 - a_4^2}{2a_3 a_4} = (s_3^2 + c_3^2) c_4 \quad (3.29)$$

$$\frac{k_1^2 + k_2^2 - a_3^2 - a_4^2}{2a_3 a_4} = c_4 \quad (3.30)$$

$$\sin \theta_4 = \pm \sqrt{1 - \cos \theta_4} \quad (3.31)$$

Sehingga untuk mencari nilai  $\theta_4$  akan menjadi sebagai berikut :

$$\theta_4 = A \tan(\sin \theta_4, \cos \theta_4) \quad (3.32)$$

Persamaan (3.25) dan (3.26) dibuat persamaan sebagai berikut agar dapat dieliminasi untuk mencari nilai  $\theta_2$

$$k_1 = a_3 c_3 + a_4 (c_4 s_3 - s_4 c_3) \quad (3.33)$$

$$k_2 = a_3 s_3 + a_4 (c_4 c_3 - s_4 s_3) \quad (3.34)$$

Dari persamaan (3.28) dan (3.29) maka dilakukan proses eliminasi untuk mencari nilai  $\theta_3$  menjadi sebagai berikut :

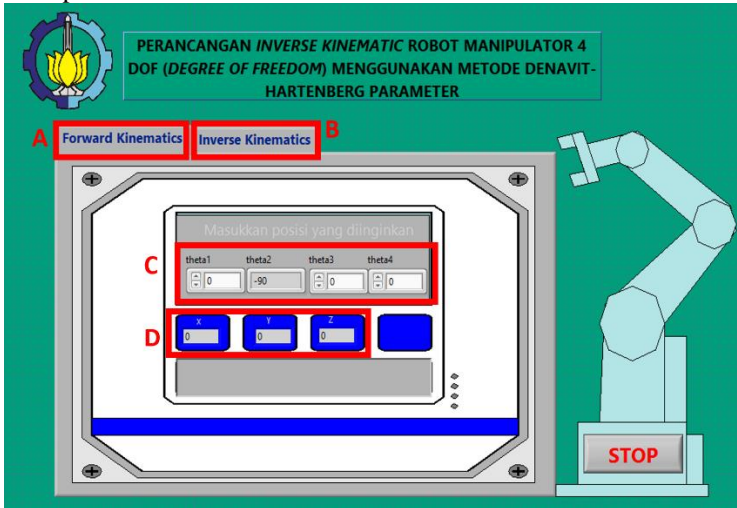
$$\cos \theta_3 = \frac{-(k_1(a_4 s_4) + k_2(a_3 + a_4 c_4))}{a_3^2 + a_4^2 + 2a_3 a_4 c_4} \quad (3.35)$$

$$\sin \theta_3 = \frac{k_1(a_3 + a_4 c_4) - k_2(a_4 s_4)}{a_3^2 + a_4^2 + 2a_3 a_4 c_4} \quad (3.36)$$

$$\theta_3 = A \tan 2(\sin \theta_3, \cos \theta_3) \quad (3.37)$$

### 3.7 Perencanaan Software

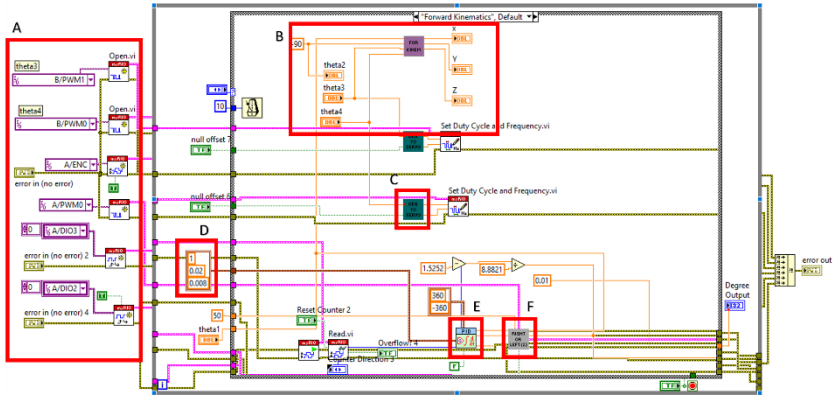
Untuk mempermudah pengaplikasian robot *manipulator* maka disusunlah sebuah *Human Machine Interface (HMI)*. *HMI* ini berisi beberapa perintah untuk menentukan arah pergerakan robot. Baik melalui metode kinematika maju maupun metode kinematika mundur.



**Gambar 3.7** Perencanaan *Human Machine Interface* (HMI)

Pada Gambar 3.7 merupakan rancangan HMI untuk mengontrol posisi robot *manipulator*. Kotak A merupakan *dialog box* dengan menggunakan analisa *forward kinematics*. Sedangkan kotak B merupakan *dialog box* dengan menggunakan analisa *inverse kinematics*. Kotak C merupakan masukkan berupa sudut untuk mengontrol ppergerakan robot *manipulator*. Sedangkan

*Diagram block* dari *human machine interface* dari robot *manipulator* seperti yang ada pada Gambar 3.8. Kotak A merupakan tempat untuk melakukan inisialisasi pin pada NI myRIO. Kemudian kotak B adalah program untuk melakukan analisa *kinematics* baik menggunakan *forward kinematics* maupun menggunakan *inverse kinematics*. Untuk lebih jelasnya untuk kotak B, C, dan F seperti pada Gambar 3.9, 3.10, dan 3.11.



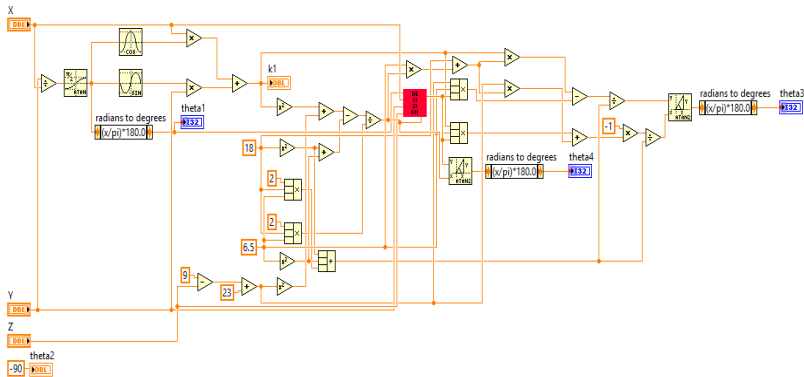
**Gambar 3.8** Diagram block Human Machine Interface pada front panel forward kinematics



**Gambar 3.9** Diagram block rumus forward kinematics

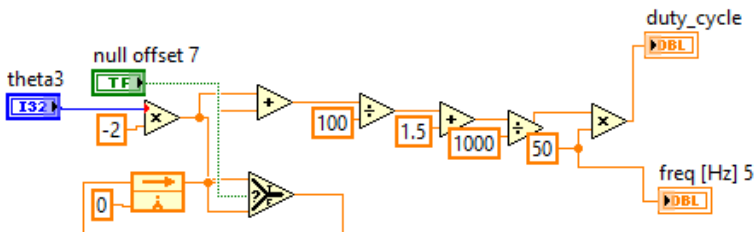
Penjabaran untuk subVI *forward kinematics* berisi program yang ada pada Gambar 3.9, dimana masukannya berupa nilai sudut yang diinginkan yang kemudian diubah ke dalam bentuk derajat dan akan dimasukkan dalam rumus *forward kinematics* seperti yang terdapat dalam Persamaan (3.9). Sedangkan keluarannya berupa posisi akhir dari *end effector*.

Penjabaran untuk subVI *inverse kinematics* berisi program yang ada pada Gambar 3.10, dimana masukannya berupa posisi yang diinginkan yang kemudian diubah ke dalam bentuk derajat dan akan dimasukkan dalam rumus *inverse kinematics* seperti yang terdapat dalam Persamaan (3.22), (3.35), dan (3.41).



**Gambar 3.10** Diagram block rumus inverse kinematics

Motor servo digunakan sebagai aktuator pada *joint elbow*, *wrist*, dan *end effector* dimana untuk menggerakannya pada posisi yang diinginkan maka perlu mengatur nilai *duty cycle*. Penjabaran untuk subVI *degree to servo* berisi program yang ada pada Gambar 3.11, dimana masukannya berupa nilai sudut atau posisi yang diinginkan yang kemudian diubah ke dalam bentuk *duty cycle*. Masukkan berupa derajat diolah menggunakan rumus konversi yang didapatkan dari data percobaan pada motor servo.



**Gambar 3.11** Konversi dari *degree* ke *duty cycle*

## BAB 4

### PENGUJIAN DAN ANALISA DATA

Pengujian dan analisis pada bab ini dilakukan untuk mengetahui performa sistem. Diawali dengan pengujian motor servo yang berfungsi sebagai *joint* pada robot *manipulator*. Kemudian pengujian metode *forward kinematics* dan *inverse kinematics* yang dilakukan dengan menyesuaikan hasil perhitungan, hasil simulasi, serta *real experiment*.

#### 4.1 Pengujian Robot Manipulator Menggunakan Metode Forward Kinematics

Metode *forward kinematics* pada pergerakan robot *manipulator* dimaksudkan untuk mengetahui posisi akhir *endeffector* setelah mendapat masukan berupa sudut.

Pada penelitian ini dilakukan pengujian pada robot *manipulator* menggunakan analisa *forward kinematics* yang telah dirancang pada Bab 3. Pengujian dilakukan dengan melakukan beberapa cara antara lain:

1. Pengujian menggunakan perhitungan manual
2. Pengujian menggunakan simulasi
3. Pengujian berdasarkan *real plant*

##### 4.1.1 Perhitungan Manual

Pada perhitungan manual menggunakan metode *forward kinematics* yang telah didapatkan sebelumnya pada Persamaan (3.8). Robot *manipulator* yang digunakan memiliki 4 buah *joint* dengan tipe *revolute* sehingga nilai  $\theta$  berupa variabel. Kemudian diberi masukan berupa nilai sudut pada masing-masing *joint*. Nilai  $\theta$  yang diberikan berbeda-beda sebanyak 3 kali pengujian.

A. Case I

Pada case pertama dimasukan nilai parameter sebagai berikut:

$$\theta_1=60^0; \theta_2=-90^0; \theta_3=30^0; \theta_4=90^0$$

Setelah didapatkan nilai parameter tersebut, maka dimasukan ke dalam persamaan (3.9) sehingga posisi *end effector* adalah:

$$\begin{bmatrix} px \\ py \\ pz \\ 1 \end{bmatrix} = \begin{bmatrix} a_2c_1c_2 + a_4c_4c_1c_{23} + a_4s_4c_1s_{23} + a_3c_1c_{23} \\ a_2s_1c_2 + a_4c_4s_1c_{23} + a_4s_4s_1s_{23} + a_3s_1c_{23} \\ d_1 - a_2s_2 - a_3s_{23} - a_4c_4s_{23} + a_4s_4c_{23} \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} px \\ py \\ pz \\ 1 \end{bmatrix} = \begin{bmatrix} 23 \cos(60) \cos(-90) + 6.5 \cos(90) \cos(60) \cos(-120) + 6.5 \sin(90) \cos(60) \sin(-120) + 18 \cos(60) \cos(-120) \\ 23 \sin(60) \cos(-90) + 6.5 \cos(90) \sin(60) \cos(-120) + 6.5 \sin(90) \sin(60) \sin(-120) + 18 \sin(60) \cos(-120) \\ 9 - 23 \sin(-90) - 18 \sin(-120) - 6.5 \cos(90) \sin(-120) + 6.5 \sin(90) \cos(-120) \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} px \\ py \\ pz \\ 1 \end{bmatrix} = \begin{bmatrix} 0 + 0 + (-4.8750) + (-4.5) \\ 0 + 0 + \\ 9 - (-23) - (-15.5885) - 0 + (-3.25) \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} px \\ py \\ pz \\ 1 \end{bmatrix} = \begin{bmatrix} 1.68542 \\ 2.91923 \\ 50.8385 \\ 1 \end{bmatrix}$$

### B. Case 2

$$\theta_1=90^0; \theta_2=-90^0; \theta_3=0^0; \theta_4=-45^0$$

Setelah didapatkan nilai parameter tersebut, maka dimasukkan ke dalam persamaan (3.9) sehingga posisi *end effector* adalah:

$$\begin{bmatrix} px \\ py \\ pz \\ 1 \end{bmatrix} = \begin{bmatrix} a_2 c_1 c_2 + a_4 c_4 c_1 c_{23} + a_4 s_4 c_1 s_{23} + a_3 c_1 c_{23} \\ a_2 s_1 c_2 + a_4 c_4 s_1 c_{23} + a_4 s_4 s_1 s_{23} + a_3 s_1 c_{23} \\ d_1 - a_2 s_2 - a_3 s_{23} - a_4 c_4 s_{23} + a_4 s_4 c_{23} \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} px \\ py \\ pz \\ 1 \end{bmatrix} = \begin{bmatrix} 23 \cos(90) \cos(-90) + 6.5 \cos(-45) \cos(90) \cos(-90) + 6.5 \sin(-45) \cos(90) \sin(-90) + 18 \cos(90) \cos(-90) \\ 23 \sin(90) \cos(-90) + 6.5 \cos(-45) \sin(90) \cos(-90) + 6.5 \sin(-45) \sin(90) \sin(-90) + 18 \sin(90) \cos(-90) \\ 9 - 23 \sin(-90) - 18 \sin(-90) - 6.5 \cos(-45) \sin(-90) + 6.5 \sin(-45) \cos(-90) \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} px \\ py \\ pz \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 4.5962 \\ 54.5962 \\ 1 \end{bmatrix}$$

### C. Case 3

$$\theta_1=45^0; \theta_2=-90^0; \theta_3=-30^0; \theta_4=90^0$$

Setelah didapatkan nilai parameter tersebut, maka dimasukkan ke dalam persamaan (3.9) sehingga posisi *end effector* adalah:

$$\begin{bmatrix} px \\ py \\ pz \\ 1 \end{bmatrix} = \begin{bmatrix} a_2 c_1 c_2 + a_4 c_4 c_1 c_{23} + a_4 s_4 c_1 s_{23} + a_3 c_1 c_{23} \\ a_2 s_1 c_2 + a_4 c_4 s_1 c_{23} + a_4 s_4 s_1 s_{23} + a_3 s_1 c_{23} \\ d_1 - a_2 s_2 - a_3 s_{23} - a_4 c_4 s_{23} + a_4 s_4 c_{23} \\ 1 \end{bmatrix}$$



$$\begin{bmatrix} px \\ py \\ pz \\ 1 \end{bmatrix} = \begin{bmatrix} 23 \cos(45) \cos(-90) + 6.5 \cos(90) \cos(45) \cos(-60) + 6.5 \sin(90) \cos(45) \sin(-60) + 18 \cos(45) \cos(-60) \\ 23 \sin(45) \cos(-90) + 6.5 \cos(90) \sin(45) \cos(-60) + 6.5 \sin(90) \sin(90) \sin(-60) + 18 \sin(45) \cos(-60) \\ 9 - 23 \sin(-90) - 18 \sin(-60) - 6.5 \cos(90) \sin(-60) + 6.5 \sin(90) \cos(-60) \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} px \\ py \\ pz \\ 1 \end{bmatrix} = \begin{bmatrix} 2.3835 \\ 2.3835 \\ 50.8385 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} px \\ py \\ pz \\ 1 \end{bmatrix} = \begin{bmatrix} 2.3835 \\ 2.3835 \\ 50.8385 \\ 1 \end{bmatrix}$$

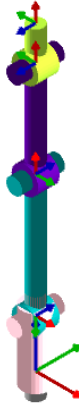
#### 4.1.2 Simulasi RoboAnalyzer

Simulasi yang digunakan adalah *software* RoboAnalyzer, dimana pada simulator ini terdapat hasil pergerakan robot menggunakan analisa *forward kinematics*.

**Tabel 4.1** Denavit-Haternberg parameter robot *manipulator* pada simulasi RoboAnalyzer

No	Link	$d_i$	$a_i$	$\alpha_i$	$\theta_i$
1	1 (Revolute)	0.09	0	-90	0
2	2 (Revolute)	0	0.23	0	-90
3	3 (Revolute)	0	0.18	180	0
4	4 (Revolute)	0	0.065	90	0

Dari Tabel 4.1 menunjukkan *Denavit Haternberg*-Parameter robot *manipulator* pada simulasi. Untuk membuat simulator, perangkat lunak yang digunakan adalah RoboAnalyzer. Karena terbatasnya *workspace* dari simulasi robot *manipulator*, maka dari itu besarnya nilai parameter  $d$  dan  $a$  diskalakan 100 kali lebih kecil dari *real plant*.



**Gambar 4.1** Posisi awal robot *manipulator* pada simulasi RoboAnalyzer

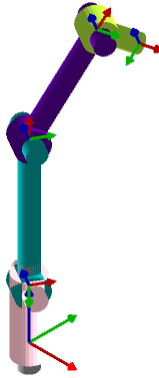
Gambar 4.1 merupakan posisi awal dari robot *manipulator* berdasarkan *Denavit Haternberg*-Parameter yang dimiliki robot *manipulator*. Garis warna biru merepresentasikan koordinat sumbu Z, garis warna merah merepresentasikan koordinat sumbu X, dan garis warna hijau merepresentasikan koordinat sumbu Y.

#### A. Case I

Pada contoh nomer 1, nilai  $\theta_1 = 60$ ,  $\theta_2 = -90$ ,  $\theta_3 = 30$  dan  $\theta_4 = 90$

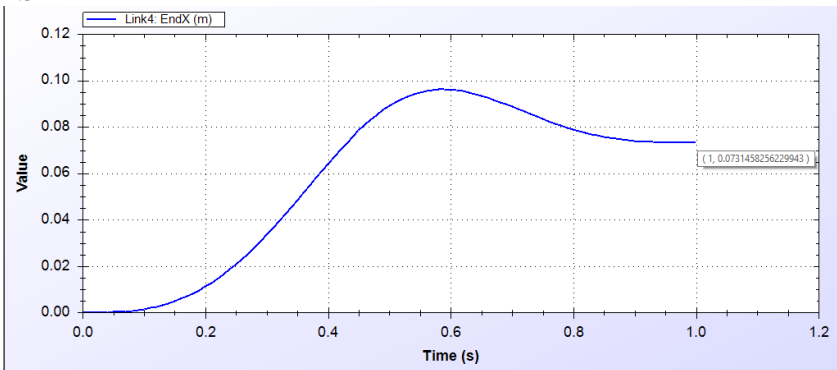
**Tabel 4.2** *Denavit-Haternberg* parameter pada case 1

No	Link	$d_i$	$a_i$	$\alpha_i$	$\theta_i$
1	1 (Revolute)	0.09	0	-90	60
2	2 (Revolute)	0	0.23	0	-90
3	3 (Revolute)	0	0.18	180	30
4	4 (Revolute)	0	0.065	90	90



**Gambar 4.2** Posisi robot *manipulator* dengan nilai  $\theta_1=60^\circ$ ;  $\theta_2=-90^\circ$ ,  $\theta_3=30^\circ$   $\theta_4=90^\circ$  pada simulasi RoboAnalyzer

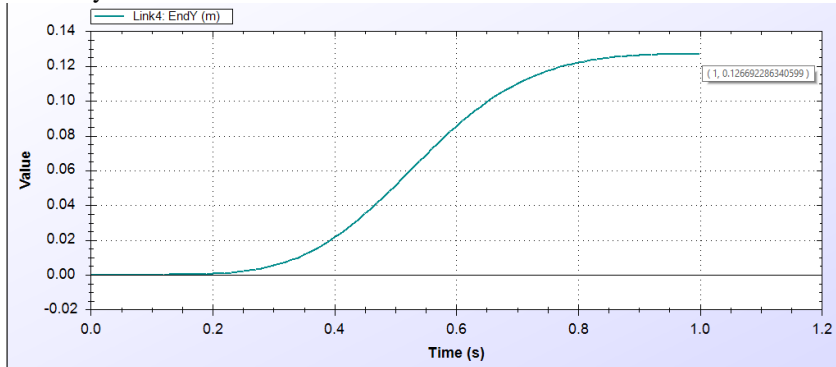
Gambar 4.2 merupakan hasil simulasi untuk arah pergerakan robot jika nilai parameter dan sudut *joint* sudah diberikan. Untuk mengetahui letak *end effector* maka dari hasil simulasi, nilai koordinat X dapat dilihat pada Gambar 4.3 . Koordinat Y sama seperti Gambar 4.4. Koordinat Z sama seperti Gambar 4.5



**Gambar 4.3** Posisi *end effector* pada koordinat x

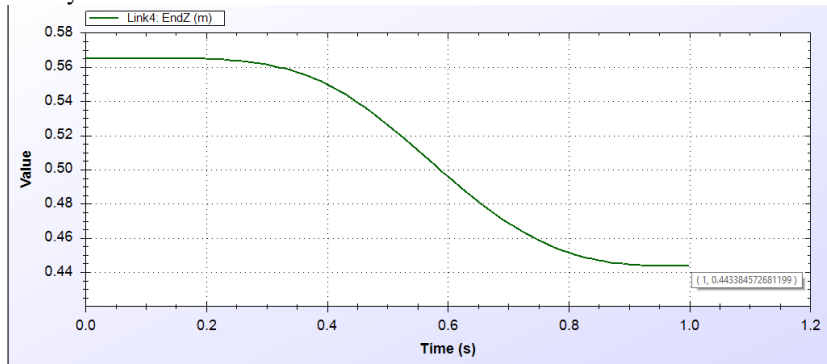
Gambar 4.3 merupakan gambar posisi *end effector* pada sumbu X. Dimana posisi awal *end effector* ada di koordinat 0, kemudian mengalami rotasi sebesar  $0^\circ$  pada *base*,  $-90^\circ$  pada *shoulder*,  $45^\circ$  pada *elbow*, dan  $90^\circ$  pada *wrist* sehingga posisi *end effector* pun berubah sedangkan waktu simulasi

adalah 1 sekon sehingga didapatkan nilai akhir posisi *end effector* pada sumbu X nilainya adalah 8.1317cm.



**Gambar 4.4** Posisi *end effector* pada koordinat y

Gambar 4.4 merupakan gambar posisi *end effector* pada sumbu Y. Dimana posisi awal *end effector* ada di koordinat 0, kemudian mengalami rotasi sebesar  $0^0$  pada *base*,  $-90^0$  pada *shoulder*,  $45^0$  pada *elbow*, dan  $90^0$  pada *wrist* sedangkan waktu simulasi 1 sekon sehingga didapatkan nilai akhir posisi *end effector* pada sumbu Y nilainya adalah 1.89E3 cm atau dapat dikatakan nilainya adalah 0cm.



**Gambar 4.5** Posisi *end effector* pada koordinat z

Gambar 4.5 merupakan gambar posisi *end effector* pada sumbu Z. Dimana posisi awal *end effector* ada di koordinat 56.5 cm, kemudian mengalami rotasi sebesar  $0^0$  pada *base*,  $-90^0$  pada *shoulder*,  $45^0$  pada *elbow*, dan  $90^0$  pada *wrist* Waktu simulasi 1 sekon sehingga didapatkan nilai akhir

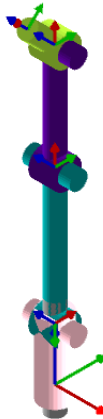
posisi *end effector* pada sumbu Y nilainya adalah 1.89E3 cm atau dapat dikatakan nilainya adalah 0cm.

#### B. Case II

Pada contoh nomer 1, nilai  $\theta_1 = 90$ ,  $\theta_2 = -90$ ,  $\theta_3 = 0$  dan  $\theta_4 = -45$

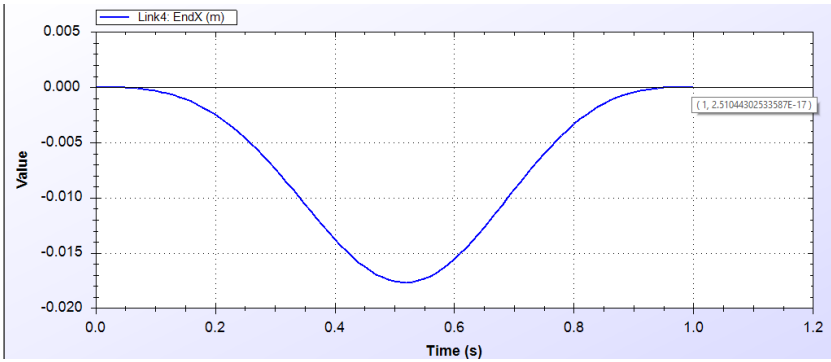
**Tabel 4.3** Denavit-Hatemberg parameter pada case 2

No	Link	$d_i$	$a_i$	$\alpha_i$	$\theta_i$
1	1 (Revolute)	0.09	0	-90	90
2	2 (Revolute)	0	0.23	0	-90
3	3 (Revolute)	0	0.18	180	0
4	4 (Revolute)	0	0.065	90	-45



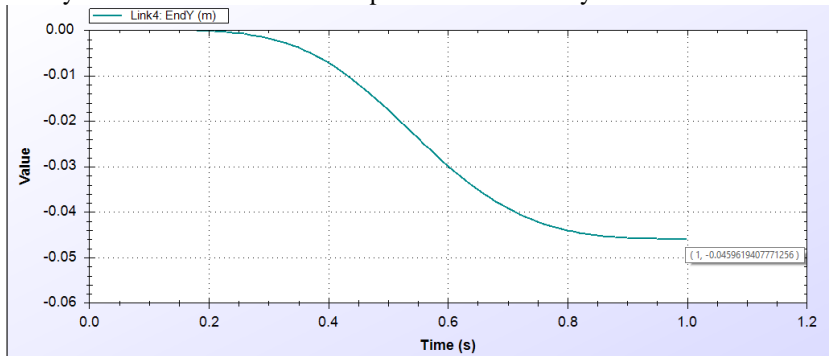
**Gambar 4.6** Posisi robot *manipulator* dengan nilai  $\theta_1=90^0$ ,  $\theta_2=-90^0$ ,  $\theta_3=0^0$ ;  $\theta_4=45^0$  pada simulasi RoboAnalyzer

Gambar 4.6 merupakan hasil simulasi untuk arah pergerakan robot jika nilai parameter dan sudut *joint* sudah diberikan. Untuk mengetahui letak *end effector* maka dari hasil simulasi, nilai koordinat x dapat dilihat pada Gambar 4.7. Koordinat Y sama seperti Gambar 4.8. Koordinat Z sama seperti Gambar 4.9



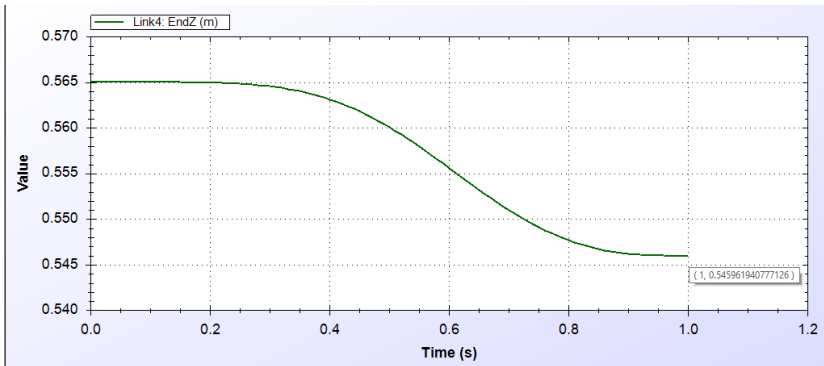
**Gambar 4.7** Posisi *end effector* pada koordinat x

Gambar 4.7 merupakan gambar posisi *end effector*. Waktu simulasi 1 sekon sehingga didapatkan nilai akhir posisi *end effector* pada sumbu X nilainya adalah 1.89E3 cm atau dapat dikatakan nilainya adalah 0cm.



**Gambar 4.8** Posisi *end effector* pada koordinat y

Gambar 4.8 merupakan gambar posisi *end effector*. Waktu simulasi 1 sekon sehingga didapatkan nilai akhir posisi *end effector* pada sumbu Y nilainya adalah 1.89E3 cm atau dapat dikatakan nilainya adalah -4.5962cm.



**Gambar 4.9** Posisi *end effector* pada koordinat z

Gambar 4.9 merupakan gambar posisi *end effector*. Waktu simulasi 1 sekon sehingga didapatkan nilai akhir posisi *end effector* pada sumbu Z nilainya adalah 1.89E3 cm atau dapat dikatakan nilainya adalah 54.5962cm.

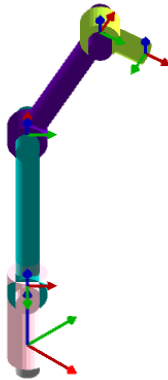
### C. Case III

Pada contoh nomer 1, nilai  $\theta_1 = 45^0$ ,  $\theta_2 = -90^0$ ,  $\theta_3 = 30^0$  dan  $\theta_4 = 90^0$

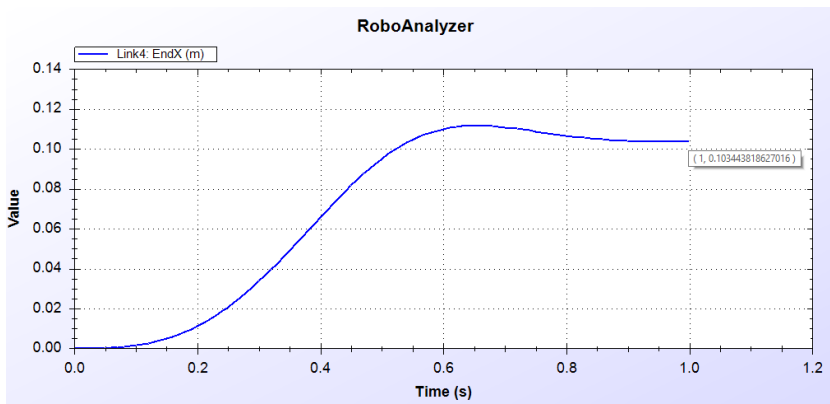
**Tabel 4.4** Denavit-Haternberg parameter pada case 3

No	Link	$d_i$	$a_i$	$\alpha_i$	$\theta_i$
1	1 (Revolute)	0.09	0	-90	45
2	2 (Revolute)	0	0.23	0	-90
3	3 (Revolute)	0	0.18	180	30
4	4 (Revolute)	0	0.065	90	90

Gambar 4.10 merupakan hasil simulasi untuk arah pergerakan robot jika nilai parameter dan sudut *joint* sudah diberikan. Untuk mengetahui letak *end effector* maka dari hasil simulasi, nilai koordinat x dapat dilihat pada Gambar 4.11. Koordinat Y seperti Gambar 4.12. Koordinat Z sama seperti Gambar 4.13.



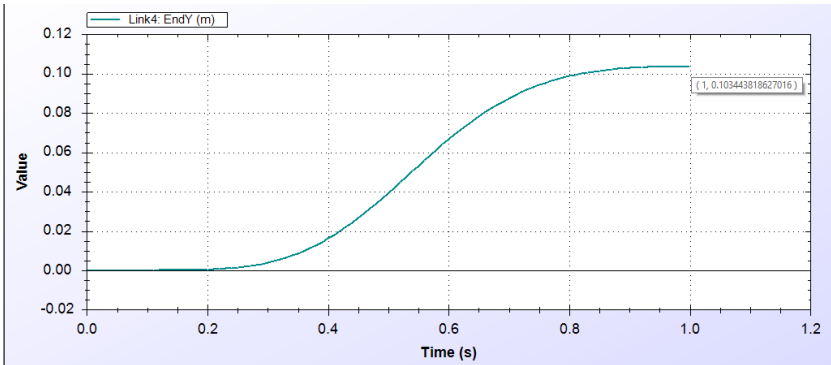
**Gambar 4.10** Posisi robot *manipulator* dengan nilai  $\theta_1=60^\circ$ ,  $\theta_2=-90^\circ$ ,  $\theta_3=90^\circ$ ,  $\theta_4=45^\circ$  pada simulasi RoboAnalyzer



**Gambar 4.11** Posisi *end effector* pada koordinat x

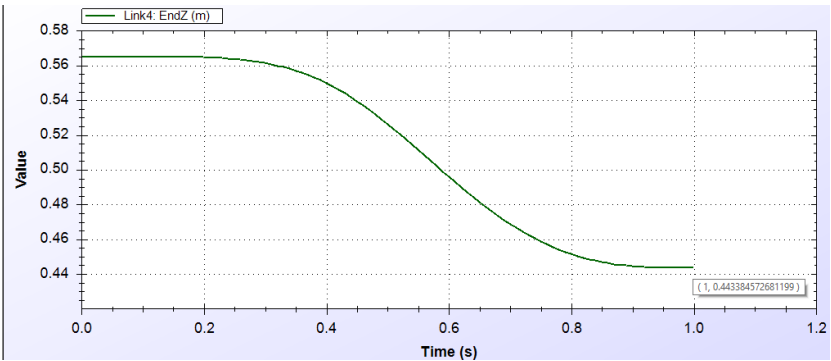
Gambar 4.11 merupakan gambar posisi *end effector*. Waktu simulasi 1 sekon sehingga didapatkan nilai akhir posisi *end effector* pada sumbu X nilainya adalah 2.3835 cm atau dapat dikatakan nilainya adalah 0cm.





**Gambar 4.12** Posisi *end effector* pada koordinat y

Gambar 4.12 merupakan gambar posisi *end effector*. Waktu simulasi 1 sekon sehingga didapatkan nilai akhir posisi *end effector* pada sumbu Y nilainya adalah 1.89E3 cm atau dapat dikatakan nilainya adalah 2.3835 cm.



**Gambar 4.13** Posisi *end effector* pada koordinat z

Gambar 4.13 merupakan gambar posisi *end effector*. Waktu simulasi 1 sekon sehingga didapatkan nilai akhir posisi *end effector* pada sumbu Z nilainya adalah 50.8385cm.

Posisi akhir *end effector* telah diketahui dari hasil grafik pada RoboAnalyzer, grafik yang bergerak secara fluktuatif menandakan koordinat awal robot *manipulator* kemudian menuju ke koordinat akhir *end effector*. Pergerakan robot *manipulator* adalah 1 sekon, sehingga pada grafik nilai

sumbu X maksimal adalah 1, hal itu merepresentasikan waktu maksimal yang perlu ditempuh robot *manipulator* bergerak ke posisi yang diinginkan.

#### 4.1.3 Real Plant

Pengujian secara *real plant* dilakukan dengan cara memberi masukan pada robot berupa sudut (*degree*) pada masing-masing *joint*. Kemudian mengukur posisi *end effector* terhadap base.

##### A. Case 1

Pengujian yang pertama dilakukan adalah memberi nilai sudut *joint* dengan nilai  $\theta_1 = 60^\circ$ ,  $\theta_2 = -90^\circ$ ,  $\theta_3 = 30^\circ$ , dan  $\theta_4 = 90^\circ$ . Dari hasil pengujian, Gambar 4.14 merupakan *real experiment* untuk *case 1*, *joint base* diberi sudut sebesar  $60^\circ$ , *joint shoulder* diberi sudut sebesar  $90^\circ$ , *joint elbow* sebesar  $30^\circ$ , dan  $90^\circ$  pada *joint wrist*. Untuk menghitung besarnya nilai posisi akhir pada sumbu x, maka dapat dilakukan pengukuran menggunakan penggaris dari *end effector* terhadap *base*.



**Gambar 4.14** *Real experiment* menggunakan metode *forward kinematics* (*case 1*)

Hasil pengukuran didapatkan posisi *end effector* pada koordinat X adalah 6.6cm, posisi *end effector* pada koordinat Y adalah 13.7cm. Sedangkan posisi *end effector* pada koordinat Z adalah 45.6cm.

**Tabel 4.5** Pengujian *endeffector* terhadap perubahan sudut *joint* menggunakan metode *forward kinematics (case 1)*

Hasil Perhitungan (cm)	Hasil Simulasi (cm)	Hasil Pengukuran (cm)	Kesalahan (%)
X=7.3146	X=7.3146	X=6.6	9.769
Y=12.6692	Y=12.6692	Y=13.7	8.136
Z=44.3385	Z=44.3385	Z=45.6	2.74

Tabel 4.5 menunjukkan persentase nilai kesalahan posisi akhir *endeffector* ketika diberi nilai  $\theta_1 = 60^\circ$ ,  $\theta_2 = -90^\circ$ ,  $\theta_3 = 30^\circ$ , dan  $\theta_4 = 90^\circ$  dan didapatkan dari hasil pengukuran yang dibandingkan dengan hasil perhitungan.

#### B. Case 2

Pengujian yang pertama dilakukan adalah memberi nilai sudut joint dengan nilai :  $\theta_1 = 90^\circ$ ,  $\theta_2 = -90^\circ$ ,  $\theta_3 = 0^\circ$ , dan  $\theta_4 = 45^\circ$ . Dari hasil pengujian, Gambar 4.15 merupakan *real experiment* untuk *case 2*, *joint base* diberi sudut sebesar  $60^\circ$ , *joint shoulder* diberi sudut sebesar  $90^\circ$ , *joint elbow* sebesar  $30^\circ$ , dan  $90^\circ$  pada *joint wrist*. Untuk menghitung besarnya nilai posisi akhir pada sumbu x, maka dapat dilakukan pengukuran menggunakan penggaris dari *end effector* terhadap *base*.



**Gambar 4.15** *Real experiment* menggunakan metode *forward kinematics (case 2)*

Hasil pengukuran didapatkan posisi *end effector* pada koordinat X adalah 0 cm, posisi *end effector* pada koordinat Y adalah 4.7cm. Sedangkan posisi *end effector* pada koordinat Z adalah 52.9 cm.

**Tabel 4.6** Pengujian *endeffector* terhadap perubahan sudut *joint* menggunakan metode *forward kinematics (case 2)*

Hasil Perhitungan (cm)	Hasil Simulasi (cm)	Hasil Pengukuran (cm)	Kesalahan (%)
X=0	X=0	X=0	0
Y=4.5962	Y=4.5962	Y=4.7	4.445
Z=54.5962	Z=54.5962	Z=52.9	3.106

Tabel 4.6 menunjukkan persentase nilai kesalahan posisi akhir *endeffector* ketika diberi nilai  $\theta_1=90^0$ ,  $\theta_2=-90^0$ ,  $\theta_3=0^0$ , dan  $\theta_4=45^0$  dan didapatkan dari hasil pengukuran yang dibandingkan dengan hasil perhitungan.

### C. Case 3

Pengujian yang pertama dilakukan adalah memberi nilai sudut joint dengan nilai :  $\theta_1=45^0$ ,  $\theta_2=-90^0$ ,  $\theta_3=30^0$ , dan  $\theta_4=90^0$ . Gambar 4.16 merupakan real experiment untuk case I, joint base diberi sudut sebesar 600, joint *shoulder* diberi sudut sebesar  $90^0$ , *joint elbow* sebesar  $30^0$ , dan  $90^0$  pada *joint wrist*. Untuk menghitung besarnya nilai posisi akhir pada sumbu x, maka dapat dilakukan pengukuran menggunakan penhharis dari *end effector* terhadap *base*.



**Gambar 4.16** *Real experiment* menggunakan metode *forward kinematics* (case 3)

Hasil pengukuran didapatkan posisi *end effector* pada koordinat X adalah 11.4 cm, posisi *end effector* pada koordinat Y adalah 10.4 cm. Sedangkan posisi *end effector* pada koordinat Z adalah 46.9 cm.

**Tabel 4.7** Pengujian *endeffector* terhadap perubahan sudut *joint* menggunakan metode *forward kinematics* (case 3)

Hasil Perhitungan (cm)	Hasil Simulasi (cm)	Hasil Pengukuran (cm)	Kesalahan (%)
X=10.3344	X=10.3344	X=11.4	10.311
Y=10.3344	Y=10.3344	Y=10.4	0.6347
Z=44.3385	Z=44.3385	Z=46.9	5.7771

Tabel 4.7 menunjukkan persentase nilai kesalahan posisi akhir *endeffector* ketika diberi nilai  $\theta_1=45^0$ ,  $\theta_2=-90^0$ ,  $\theta_3=30^0$ , dan  $\theta_4=90^0$  dan didapatkan dari hasil pengukuran yang dibandingkan dengan hasil perhitungan.

#### 4.1.4 Persentase Nilai Kesalahan Berdasarkan Hasil Perhitungan dengan Hasil Pengukuran

Table 4.5 hingga Tabel 4.7 menampilkan hasil posisi akhir *endeffector* ketika diberi sudut yang berbeda. Berdasarkan hasil pengujian *endeffector* menggunakan metode *forward kinematics* baik menggunakan perhitungan,

simulasi, dan pengukuran secara *real plant* maka diketahui nilai kesalahan untuk koordinat X adalah 6.693%, untuk koordinat Y adalah 4.4052%, dan untuk koordinat Z adalah 3.8743% seperti yang ditunjukkan pada Tabel 4.8

**Tabel 4.8** Persentase nilai kesalahan *endeffector* terhadap perubahan koordinat x, y, dan z menggunakan metode *forward kinematics*

Koordinat	Case 1 (%)	Case 2 (%)	Case 3 (%)	Kesalahan (%)
X	9.769	0	10.311	6.693
Y	8.136	4.445	0.6347	4.4052
Z	2.74	3.106	5.7771	3.8743

## 4.2 Pengujian Robot Manipulator Menggunakan Metode *Inverse Kinematics*

Metode *inverse kinematics* pada pergerakan robot *manipulator* dimaksudkan untuk mengetahui sudut pada masing-masing *joint* setelah mendapat masukan berupa posisi akhir. Melakukan perhitungan *inverse kinematics*.

Pada penelitian ini dilakukan pengujian pada robot *manipulator* menggunakan analisa *inverse kinematics* yang telah dirancang pada Bab 3. Pengujian dilakukan dengan melakukan beberapa cara pengambilan data posisi akhir *endeffector* antara lain:

1. Pengujian menggunakan perhitungan manual
2. Pengujian berdasarkan *real plant*

### 4.2.1 Perhitungan Manual

Pada perhitungan manual menggunakan metode *inverse kinematics* yang telah didapatkan sebelumnya pada Persamaan (3.20), (3.32), dan (3.37). Kemudian diberi masukan berupa nilai posisi akhir *endeffector*. Posisi akhir *endeffector* yang diberikan berbeda-beda sebanyak 3 kali pengujian.

Untuk melakukan pengujian, maka dimasukan nilai-nilai posisi dari hasil *forward kinematics* kemudian dilakukan perhitungan menggunakan *inverse kinematics* untuk menentukan sudut pada masing-masing *joint*.

#### A. Case I

Pada *case 1* ini, *end effector* bergerak menuju posisi koordinat X= 0, Y= 8.13173, Z= 49.3241 sehingga untuk menyelesaikannya menggunakan rumus dari persamaan (3.20), sehingga didapatkan nilai  $\theta_1$

$$\theta_1 = A \tan\left(\frac{y}{x}\right)$$

$$\theta_1 = A \tan\left(\frac{8.13173}{0}\right)$$

$$\theta_1 = A \tan(\infty)$$

$$\theta_1 = 90^0$$

Setelah mendapatkan nilai  $\theta_1$ , maka dapat mencari nilai  $\theta_3$  dan  $\theta_4$  dari persamaan (3.30) dan (3.31) maka

$$k_1 = 8.13173$$

$$k_2 = -17.3241$$

$$\cos \theta_4 = 0$$

$$\sin \theta_4 = \pm \sqrt{1 - \cos^2 \theta_4}$$

$$\sin \theta_4 = \pm 1$$

$$\theta_4 = A \tan\left(\frac{\sin \theta_4}{\cos \theta_4}\right) = A \tan(\infty)$$

$$\theta_4 = 90$$

Dari persamaan (3.35) dan (3.36) maka

$$\sin \theta_3 = 0.431449$$

$$\cos \theta_3 = 0.431449$$

$$\theta_3 = A \tan(\sin \theta_3, \cos \theta_3)$$

$$\theta_3 = 45$$

Nilai  $\theta_1$ ,  $\theta_2$ ,  $\theta_3$ , dan  $\theta_4$  yang telah ditemukan menjadi masukan sudut masing-masing *joint* pada robot *manipulator*.

## B. Case 2

Pada *case 2* ini, *end effector* bergerak menuju posisi koordinat  $X=13.3208$ ,  $Y=-13.3208$ ,  $Z=35.3708$  sehingga untuk menyelesaikannya menggunakan rumus dari persamaan (3.20), sehingga didapatkan nilai  $\theta_1$

$$\theta_1 = A \tan\left(\frac{y}{x}\right)$$

$$\theta_1 = A \tan\left(\frac{13.3208}{-13.3208}\right)$$

$$\theta_1 = A \tan(-1)$$

$$\theta_1 = -45$$

$$\theta_4 = A \tan(\sin \theta_4, \cos \theta_4)$$

Dari persamaan (3.30) dan (3.31) maka

$$k_1 = 18.8385$$

$$k_2 = -3.3708$$

$$\cos \theta_4 = 0$$

$$\sin \theta_4 = \pm \sqrt{1 - \cos^2 \theta_4}$$

$$\sin \theta_4 = -1$$

$$\theta_4 = A \tan\left(\frac{\sin \theta_4}{\cos \theta_1}\right) = A \tan\left(\frac{-1}{0}\right) = A \tan(-\infty)$$

$$\theta_4 = -90$$

Dari persamaan (3.35) dan (3.36) maka

$$\sin \theta_3 = 0.528416$$

$$\cos \theta_3 = 0.30508$$

$$\theta_3 = A \tan(\sin \theta_3, \cos \theta_3)$$

$$\theta_3 = 60$$

Nilai  $\theta_1$ ,  $\theta_2$ ,  $\theta_3$ , dan  $\theta_4$  yang telah ditemukan menjadi masukan sudut masing-masing *joint* pada robot *manipulator*.

### C. Case III

Pada *case 3* ini, *end effector* bergerak menuju posisi koordinat  $X=0$ ,  $Y=-8.13173$ ,  $Z=49.3241$  sehingga untuk menyelesaikannya menggunakan rumus dari persamaan (3.20), sehingga didapatkan nilai  $\theta_1$



$$\theta_1 = A \tan\left(\frac{y}{x}\right)$$

$$\theta_1 = A \tan\left(\frac{-8.13173}{0}\right)$$

$$\theta_1 = A \tan(-\infty)$$

$$\theta_1 = -90$$

$$\theta_4 = A \tan(\sin \theta_4, \cos \theta_4)$$

Dari persamaan (3.30) dan (3.31) maka

$$k_1 = 8.13173$$

$$k_2 = -17.3241$$

$$\cos \theta_4 = 0$$

$$\sin \theta_4 = \pm \sqrt{1 - \cos^2 \theta_4}$$

$$\sin \theta_4 = 1$$

$$\theta_4 = A \tan\left(\frac{\sin \theta_4}{\cos \theta_4}\right) = A \tan\left(\frac{1}{0}\right) = A \tan(\infty)$$

$$\theta_4 = 90$$

Dari persamaan (3.35) dan (3.36) maka

$$\sin \theta_3 = 0.43145$$

$$\cos \theta_3 = 0.43145$$

$$\theta_3 = A \tan(\sin \theta_3, \cos \theta_3)$$

$$\theta_3 = 45$$

Nilai  $\theta_1$ ,  $\theta_2$ ,  $\theta_3$ , dan  $\theta_4$  yang telah ditemukan menjadi masukan sudut masing-masing *joint* pada robot *manipulator*.

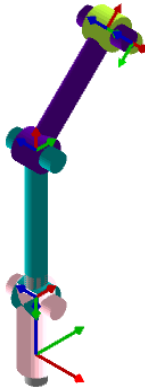
#### 4.2.2 Real Plant

Pengujian secara *real plant* dilakukan dengan cara memberi masukan berupa posisi akhir *endeffector* yang ingin dicapai. Kemudian mengukur posisi *end effector* terhadap base.

##### A. Case 1

Pada pengujian yang pertama, posisi akhir *endeffector* yang diinginkan adalah  $X = 0$ ,  $Y = 8.13173$ ,  $Z = 49.3241$ . Gambar 4.17 merupakan gambar simulasi menggunakan RoboAnalyzer untuk pengujian yang pertama. Dimana arah *endeffector* pada simulasi untuk koordinat X digambarkan dengan garis

warna merah, garis hijau menunjukkan arah koordinat Y, dan garis warna biru merepresentasikan arah koordinat Z. Sedangkan pada Gambar 4.18 merupakan *real experiment* untuk *case 1*, posisi yang harus dicapai *endeffector* adalah pada sumbu  $X=0\text{cm}$ , sumbu  $Y=8.13173\text{cm}$ , dan sumbu  $Z=49.3241\text{ cm}$ .



**Gambar 4.17** Posisi *end effector* pada robot *manipulator*



**Gambar 4.18** *Real experiment* menggunakan metode *inverse kinematics* (*case 1*)

Tabel 4.9 menunjukkan persentase nilai kesalahan posisi akhir *endeffector* berada pada koordinat  $X=0\text{cm}$ ,  $Y=8.13173\text{cm}$ , dan  $Z=49.3241\text{ cm}$  dan didapatkan dari hasil pengukuran yang dibandingkan dengan hasil perhitungan

**Tabel 4.9** Pengujian *endeffector* terhadap perubahan sudut *joint* menggunakan metode *inverse kinematics (case 1)*

Hasil perhitungan (cm)	Hasil simulasi (cm)	Hasil pengukuran (cm)	Kesalahan (%)
X=0	X=0	X=0	0
Y=8.13173	Y=8.13173	Y=7.7	5.3092
Z=49.3241	Z=49.3241	Z=45.9	6.942

#### B. Case 2

Pada pengujian yang pertama, posisi akhir *endeffector* yang diinginkan adalah  $X= 13.3208$ ,  $Y= -13.3208$ ,  $Z= 35.3708$ . Gambar 4.19 merupakan gambar simulasi menggunakan RoboAnalyzer untuk pengujian yang pertama. Dimana arah *endeffector* pada simulasi untuk koordinat X digambarkan dengan garis warna merah, garis hijau menunjukkan arah koordinat Y, dan garis warna biru merepresentasikan arah koordinat Z. Sedangkan pada gambar 4.20 merupakan real experiment untuk case 2, posisi yang harus dicapai *endeffector* adalah pada sumbu  $X=13.3208\text{cm}$ , sumbu  $Y=-13.3208\text{ cm}$ , dan sumbu  $Z=35.3708\text{cm}$ .



**Gambar 4.19** Posisi *end effector* pada robot *manipulator*



**Gambar 4.20** Real experiment menggunakan metode inverse kinematics (case 2)

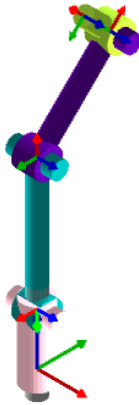
**Tabel 4.10** Pengujian *endeffector* terhadap perubahan sudut *joint* menggunakan metode inverse kinematics (case 2)

Hasil perhitungan (cm)	Hasil simulasi (cm)	Hasil pengukuran (cm)	Kesalahan (%)
X=13.3208	X=13.3208	X=11.5	13.6688
Y=-13.3208	Y=-13.3208	Y=-17.5	31.378
Z=35.3708	Z=35.3708	Z=34.15	3.4514

### C. Case 3

Pada pengujian yang pertama, posisi akhir *endeffector* yang diinginkan adalah  $X = 0$ ,  $Y = -8.13173$ ,  $Z = 49.3241$ . Gambar 4.21 merupakan gambar simulasi menggunakan RoboAnalyzer untuk pengujian yang pertama. Dimana arah *endeffector* pada simulasi untuk koordinat X digambarkan dengan garis warna merah, garis hijau menunjukkan arah koordinat Y, dan garis warna biru merepresentasikan arah koordinat Z. Sedangkan pada gambar 4.22 merupakan

*real experiment* untuk *case 3*, posisi yang harus dicapai *end effector* adalah pada sumbu  $X=0\text{cm}$ , sumbu  $Y=-8.13173\text{cm}$ , dan sumbu  $Z=49.3241\text{ cm}$ .



**Gambar 4.21** Posisi *end effector* pada robot manipulator



**Gambar 4.22** *Real experiment* menggunakan metode *inverse kinematics* (*case 3*)

**Tabel 4.11** Pengujian *endeffector* terhadap perubahan sudut *joint* menggunakan metode *inverse kinematics* (case 3)

Hasil perhitungan (cm)	Hasil simulasi (cm)	Hasil pengukuran (cm)	Kesalahan (%)
X=0	X=0	X=0	0
Y=-8.13173	Y=-8.13173	Y=-7.5	7.768
Z=49.3241	Z=49.3241	Z=45.2	8.3612

Dari data percobaan dapat dilihat nilai kesalahan yang cukup besar antara *real experiment* dengan simulasi maupun hasil perhitungan. Hal ini dikarenakan posisi *elbow* pada *manipulator* menggunakan *gear* sehingga terdapat banyak permasalahan mekanik. Pergerakan *gear* yang tidak selalu presisi dikarenakan pembebanan.

#### 4.2.3 Persentase Nilai Kesalahan Berdasarkan Hasil Perhitungan dengan Hasil Pengukuran

Table 4.9 hingga Tabel 4.11 menampilkan hasil posisi akhir *endeffector* ditugaskan untuk mencapai posisi tertentu. Berdasarkan hasil pengujian *endeffector* menggunakan metode *inverse kinematics* baik menggunakan perhitungan, simulasi, dan pengukuran secara *real plant* maka diketahui nilai kesalahan untuk koordinat X adalah 4.556%, untuk koordinat Y adalah 14.818%, dan untuk koordinat Z adalah 6.2515% seperti yang ditunjukkan pada Tabel 4.12

**Tabel 4.12** Persentase nilai kesalahan *endeffector* terhadap perubahan koordinat x, y, dan z menggunakan metode *inverse kinematic*

Koordinat	Case 1 (%)	Case 2 (%)	Case 3 (%)	Kesalahan (%)
X	0	13.6688	0	4.556
Y	5.3092	31.378	7.768	14.818
Z	6.942	3.4514	8.3612	6.2515

## **BAB 5**

### **PENUTUP**

Metode *inverse kinematics* mampu membuat pergerakan robot *manipulator* bergerak ke posisi yang diinginkan. Namun dalam penggunaannya penyelesaian *inverse kinematics* bisa berupa bilangan *infinity* untuk nilai sudut pada masing-masing *joint*. Dari hasil pengujian *inverse kinematics* didapatkan nilai kesalahan untuk koordinat X, Y, dan Z sebesar 6.693%, 4.4052%, dan 3.8743%.

Saran untuk penelitian selanjutnya adalah penggunaan metode *differential inverse kinematics* agar pergerakan robot *manipulator* lebih halus untuk mencapai ke posisi yang diinginkan



*Halaman ini sengaja dikosongkan*

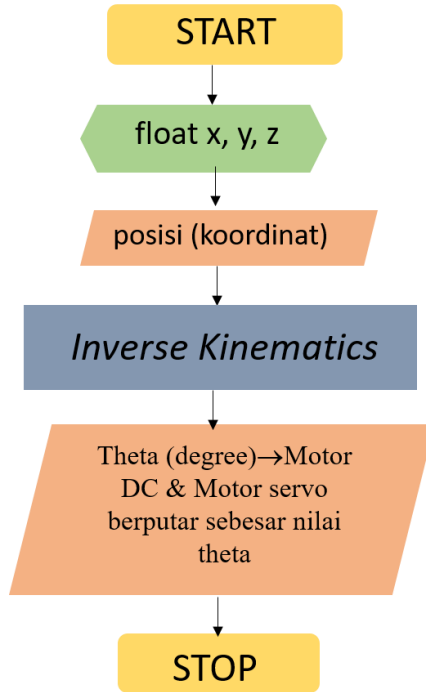
## DAFTAR PUSTAKA

- [1] E. Pitawarno, Robotika (Desain, Kontrol, dan Kecerdasan Buatan), Yogyakarta: Andi OFFSET, 2006.
- [2] I. S. Department, "Executive Summary World Robotics 2016 Industrial Robots," World Robotic, 2016.
- [3] B. S. Richard P. Paul, "Kinematic Control Equations For Simple Manipulators," in *IEEE Conference on Decision and Control including the 17th Symposium on Adaptive Processes*, 1978.
- [4] X. W. D. L. C. Jing HUANG, "A new method for solving inverse kinematics of an industrial robot," *IEEE*, vol. 3, pp. 53-56, 2012.
- [5] M. V. Mark W. Spong, Robot Dynamic Andd Control, Wiley, 2006.
- [6] A. M. B. Subudhi, "Dynamic Modelling, Simulation and Control of A Manipulator With Flexible Links and Joints," *IEEE*, p. 257–270, 2002.
- [7] L.-W. Tsai, Robot Analysis: The Mechanics of Serial and Paralel Manipulators, New York: Wiley, 1999.
- [8] \_\_\_\_\_, NI myRIO 1990 Datasheet.
- [9] F. A. Adi Saputro, Perancangan Human Machine Interface Robot Lengan Tetrix Menggunakan Berbasis Open-GL, Surabaya: ITS, 2016.
- [10] Q. J. G. C. H. S. Yang Si, "A Complete Solution to the Inverse Kinematics Problem for 4--DOF Manipulator Robot," *IEEE 8th Conference on Industrial Electronics and Applications (ICIEA)*, vol. 8, 2013.
- [11] C. Melchiorri, "Dipartimento di Ingegneria dell'Energia Elettrica e dell'Informazione (DEI)," 2014. [Online]. Available: [http://www-lar.deis.unibo.it/people/cmelchiorri/Files\\_Robotica/FIR\\_01\\_Industrial\\_Robotics.pdf](http://www-lar.deis.unibo.it/people/cmelchiorri/Files_Robotica/FIR_01_Industrial_Robotics.pdf). [Accessed 20 November 2016].



## LAMPIRAN

**Lampiran 1.** *Flowchart* Keseluruhan Sistem



Bentuk persegi merepresentasikan awal mulai program, kemudian masukkan nilai posisi *endeffector* dalam koordinat X, Y, dan Z. Nilai posisi yang diinginkan menjadi masukan untuk metode *inverse kinematics*. Keluaran *inverse kinematics* berupa nilai sudut yang harus ditempuh masing-masing *joint*, nilai sudut menjadi masukan untuk aktuator. Pada gambar *flowchart* di atas, *input* berupa posisi yang diinginkan kemudian akan diolah menggunakan rumus *inverse kinematics* sehingga dihasilkan nilai theta. Kemudian nilai theta tersebut akan menjadi masukan pada motor DC dan motor servo.

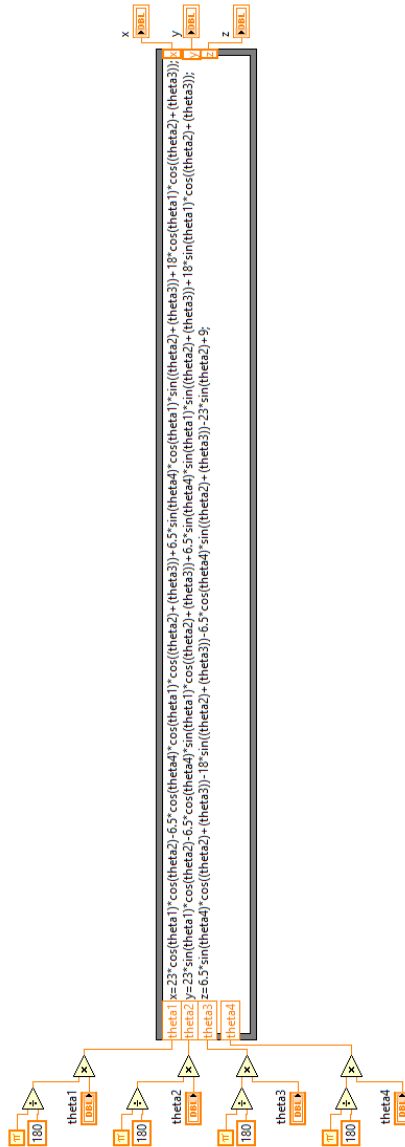
*Halaman ini sengaja dikosongkan*

## Lampiran 2. Rumus Transformasi Homogen

$$\begin{aligned}
 {}^0A_4 &= {}^0A_1 {}^1A_2 {}^2A_3 {}^3A_4 \\
 {}^0A_4 &= \begin{bmatrix} {}^0R_4 & {}^4P \\ 0_{(1 \times 3)} & 1_{(1 \times 1)} \end{bmatrix} \\
 {}^0A_4 &= \begin{bmatrix} c_4c_1c_{23} + s_4c_1s_{23} & s_1 & a_2c_1c_2 - a_4c_4c_1c_{23} + a_4s_4c_1s_{23} + a_3c_1c_{23} \\ c_4s_1c_{23} + s_4s_1s_{23} & -c_1 & a_2s_1c_2 - a_4c_4s_1c_{23} + a_4s_4s_1s_{23} + a_3s_1c_{23} \\ s_4c_{23} - c_4s_{23} & 0 & d_1 - a_2s_2 - a_3s_{23} - a_4c_4s_{23} + a_4s_4c_{23} \\ 0 & 0 & 1 \end{bmatrix}
 \end{aligned}$$

Halaman ini sengaja dikosongkan

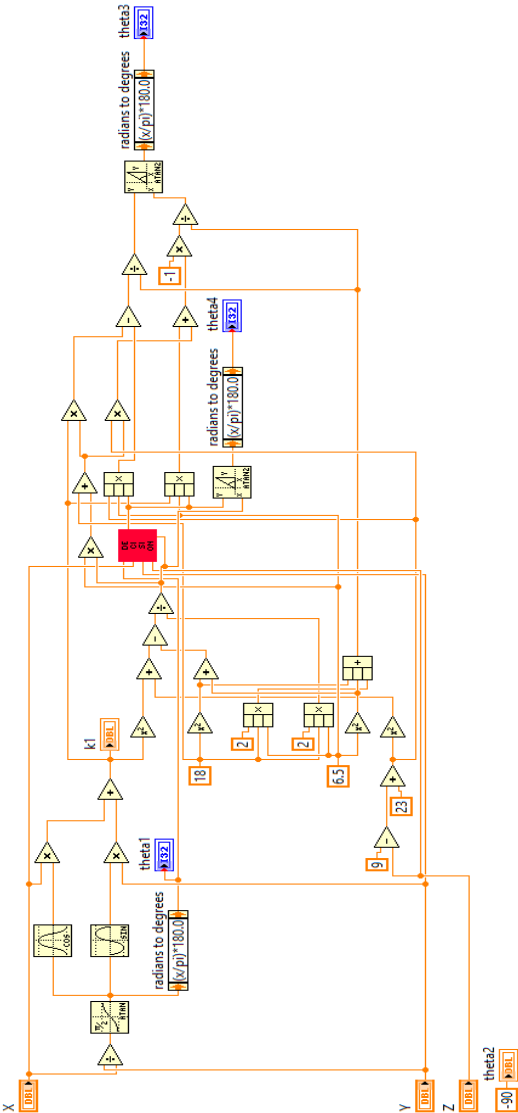
### Lampiran 3. Listing Program Metode Forward Kinematics





*Halaman ini sengaja dikosongkan*

Lampiran 4. Listing Program Metode Inverse Kinematics



*Halaman ini sengaja dikosongkan*

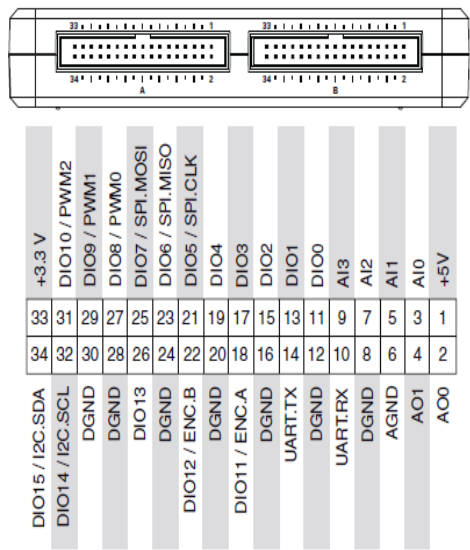
Lampiran 5. Datasheet NI myRIO

5.1 Connector Pinouts

Connector Pinouts

NI myRIO-1900 Expansion Port (MXP) connectors A and B carry identical sets of signals. The signals are distinguished in software by the connector name, as in ConnectorA/DIO1 and ConnectorB/DIO1. Refer to the software documentation for information about configuring and using signals. The following figure and table show the signals on MXP connectors A and B. Note that some pins carry secondary functions as well as primary functions.

Figure 3. Primary/Secondary Signals on MXP Connectors A and B



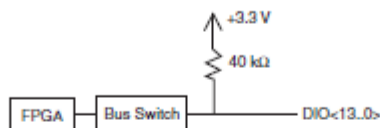
## 5.2 Digital Input Output NI myRIO

### DIO Lines

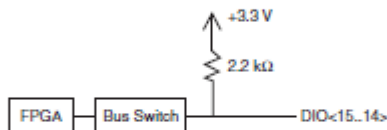
The NI myRIO-1900 has 3.3 V general-purpose DIO lines on the MXP and MSP connectors. MXP connectors A and B have 16 DIO lines per connector. On the MXP connectors, each DIO line from 0 to 13 has a 40 k $\Omega$  pullup resistor to 3.3 V, and DIO lines 14 and 15 have 2.2 k $\Omega$  pullup resistors to 3.3 V. MSP connector C has eight DIO lines. Each MSP DIO line has a 40 k $\Omega$  pulldown resistor to ground. DGND is the reference for all the DIO lines. You can program all the lines individually as inputs or outputs. Secondary digital functions include Serial Peripheral

Interface Bus (SPI), I<sup>2</sup>C, pulse-width modulation (PWM), and quadrature encoder input. Refer to the NI myRIO software documentation for information about configuring the DIO lines.

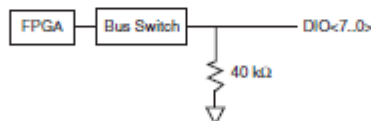
**Figure 7.** DIO Lines <13..0> on MXP Connector A or B



**Figure 8.** DIO Lines <15..14> on MXP Connector A or B



**Figure 9.** DIO Lines <7..0> on MSP Connector C



When a DIO line is floating, it floats in the direction of the pull resistor. A DIO line may be floating in any of the following conditions:

- when the myRIO device is starting up
- when the line is configured as an input
- when the myRIO device is powering down

You can add a stronger resistor to a DIO line to cause it to float in the opposite direction.

## RIWAYAT PENULIS



Adi Novitarini Putri lahir di kota Surabaya, provinsi Jawa Timur pada tanggal 14 November 1995. Penulis merupakan anak pertama dari tiga bersaudara. Setelah menyelesaikan Sekolah Dasar, Sekolah Menengah Pertama dan Sekolah Menengah Atas di daerah asalnya yaitu Surabaya pada tahun 2014, dengan izin Allah SWT serta kedua orang tua, penulis melanjutkan pendidikan di Jurusan Teknik Elektro Otomasi, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember, Surabaya.

*E-mail:* [adinovitarini@gmail.com](mailto:adinovitarini@gmail.com)

*Halaman ini sengaja dikosongkan*